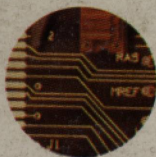
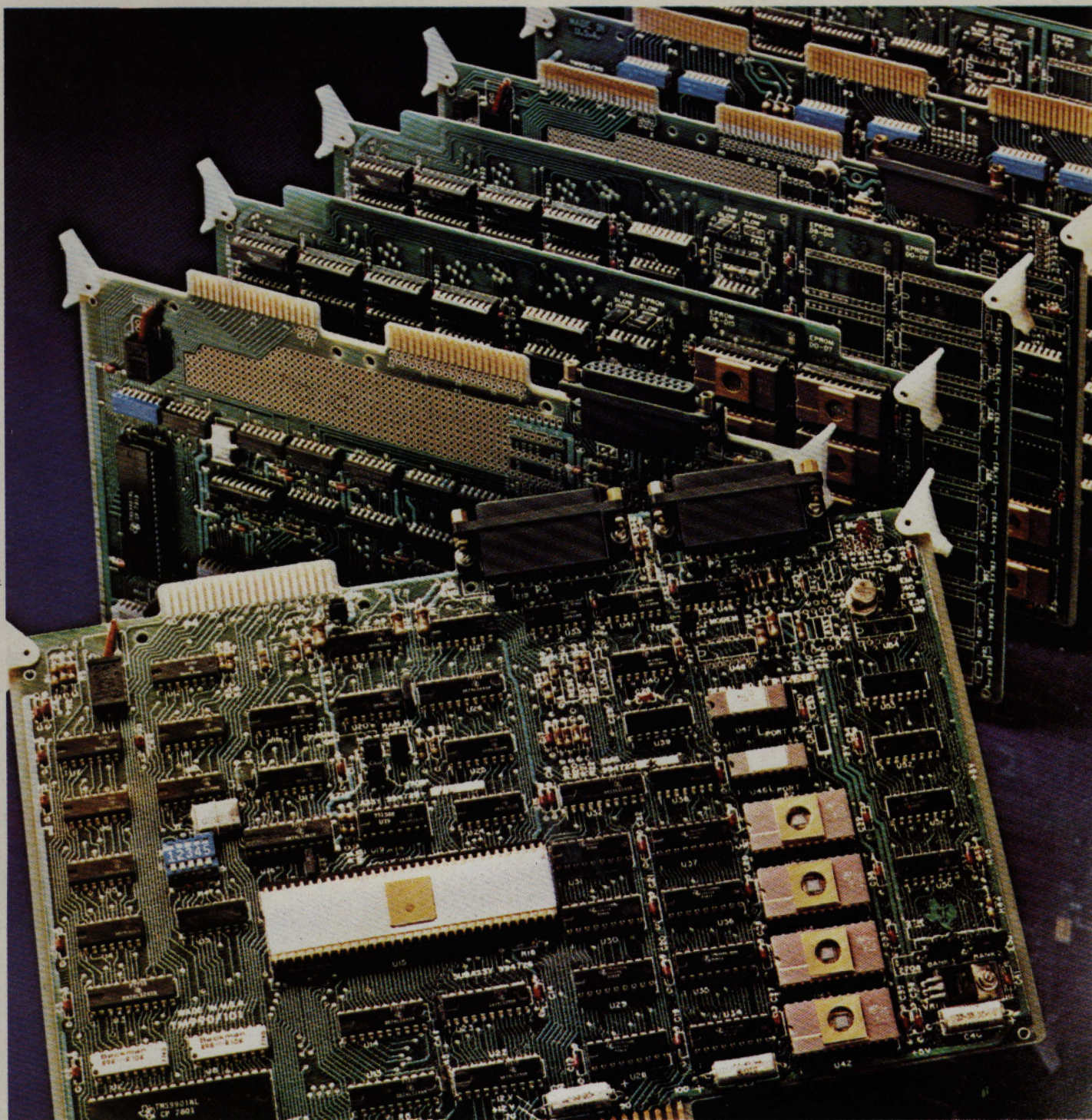
UNA PUBBLICAZIONE
DEL GRUPPO EDITORIALE JACKSON
ANNO 1 N. 1
Bit
MICROPROCESSORS - HARDWARE - SOFTWARE
HOME & PERSONAL COMPUTERS
L. 1500

La Texas Instruments annuncia il primo modulo di microcomputer a 16 BIT con linguaggio BASIC. Più facile da programmare, ampliato nella capacità di memorizzazione e di comunicazione

Sono disponibili presso i distributori autorizzati T.I., dei veri e propri computers su singola scheda.
Stiamo parlando dei nuovi moduli di microcomputers TM 990/101M.
La maggiore estensione di memoria, la possibilità di comunicazioni simultanee su due porte (con protocolli di comunicazione già implementati su board), e la disponibilità del POWER BASIC, linguaggio ad alto livello che rende estremamente veloce la programmazione, sono la caratteristiche fondamentali di questi moduli.

Questi nuovi moduli della Texas Instruments riducono i tempi di progetto e di sviluppo, il numero di

componenti del sistema ed i costi di realizzazione.

Queste unità sono preassemblate, pretestate e pronte all'uso, e questo garantisce un'alta affidabilità al sistema.

Estensione di memoria quattro volte più grande

I microcomputers TM 990/101M possiedono una capacità di memoria RAM statica su board che va da 1K a 2K parole da 16 bits.
L'estensione di EPROM va da 2K a 4K parole da 16 bits.

Capacità di comunicazione doppia

Sul board sono disponibili due porte per comunicazioni seriali: una usata per utilizzazioni "remote" (terminali o modem), l'altra per utilizzazioni "locali" (terminali EIA, teletype, microterminale TM 990/301 della T.I.).

Il meglio nell'ambito dei microcomputers

Nella serie TM 990 della T.I. troverete la più ampia scelta possibile per prestazioni/costo, di microcomputers a 16 bits che possono soddisfare le vostre necessità a livello di sistema. Ideali per valutazioni su microprocessori e per accelerare il progetto di sistemi a microprocessori, rappresentano una valida alternativa in produzione.
Caratteristica fondamentale di tutti i moduli è la completa compatibilità a livello di set di istruzioni con gli altri membri dell'intera famiglia TM 990/9900.

Per valutazioni ed applicazioni OEM

- TM 990/100M - Utilizza il microprocessore TMS 9900 a 16 bit in tecnologia N MOS, della Texas Instruments.

  1K bytes di RAM statica, 2K bytes di EPROM, e porte di I/O seriali e parallele programmabili, fanno di questa unità un potente microcomputer su singola scheda.

- TM 990/180M - Utilizza il microprocessore TMS 9980 a 16 bit, in tecnologia N MOS, della T.I., che consente di lavorare a 2,5 MHz, e che contengono una interfaccia a 8 bit per la memoria, minimizza i costi dei sistemi più piccoli.

Per espansioni di memoria

- TM 990/201 - 8K bytes di EPROM e 4K bytes di RAM statica. Espandibile a 32K bytes di EPROM e 16K bytes di RAM.
- TM 990/206 - 8K bytes di RAM statica espandibile a 16K bytes.

Per ingresso dati e monitoraggio

- TM 990/301 - Consente l'ingresso di dati di programma, la visualizzazione e la modifica del contenuto dei registri interni e della memoria sotto controllo software (TIBUG).

Per espandere le linee di I/O

- TM 990/310 - Modulo di espansione delle linee di I/O a 48 bit.

Per sviluppare software e per la produzione

- TM 990/401 - Monitor interattivo di debug (TIBUG) già programmato in EPROM.
- TM 990/402 - Assemblatore line-by-line già programmato in EPROM.
- TM 990/450 - POWER BASIC già programmato in 8K bytes di EPROM.
- TM 990/451 - POWER BASIC già programmato in 12K bytes di EPROM.

Sono inoltre disponibili una vasta gamma di connettori e cavi per gli usi più comuni, schede di prototipizzazione, ed uno chassis a 4-slot (TM 990/510) già predisposto per ospitare 4 schede della famiglia TM 990.

Supporto software

I moduli di microcomputer della serie TM 990 sono completamente supportati dal sistema di sviluppo AMPL (Advanced Microprocessor Prototyping Laboratory) della Texas Instruments.

L'AMPL consente di effettuare trace a 10 MHz ed emulazioni per i microprocessori TMS 9900, 9980, SBP 9900, TMS 9940 e per tutti i microprocessori della T.I. che saranno prodotti in futuro.

L'AMPL è disponibile nella configurazione con sistema a floppy disk o in quella con sistema a dischi rigidi in modo da soddisfare le più svariate esigenze degli utenti. I programmi possono essere editati, assemblati, lincati, caricati ed eseguiti più velocemente che con i sistemi tradizionali a nastri o a cassette.

L'emulazione del TMS 9900/9980 consente di sviluppare e di debuggare il software direttamente su un modulo TM 990 visualizzando e controllando, contemporaneamente, le operazioni per mezzo del sistema di sviluppo AMPL.

Per salvaguardare il progetto di oggi dalle innovazioni di domani

I microcomputers della serie TM 990 e l'AMPL si integrano perfettamente con l'intera famiglia 990/9900 rendendola veramente completa e di estema potenzialità nell'ambito delle applicazioni a 16 bit.

Questa famiglia è composta da una vasta gamma di microprocessori, microcomputers e minicomputers che usano tutti la stessa architettura orientata verso la memoria, lo stesso set di istruzioni; sono tutti completamente compatibili e sono previsti per essere supportati dallo stesso sistema di sviluppo.

Con la famiglia 990/9900 il progettista può scegliere la soluzione a costo più basso, e più soddisfacente alle sue esigenze, senza preoccuparsi degli sviluppi futuri del suo prodotto, in quanto la completa compatibilità software rende il progetto software direttamente utilizzabile su prodotti che si possono diversificare per sofisticazione hardware.
Per maggiori informazioni contattate i nostri distributori autorizzati oppure l'ufficio Promozione Commerciale, Cittaducale - RIETI

TEXAS INSTRUMENTS
SEMICONDUTTORI ITALIA S.p.A.
Elettronica per il progresso.

SOMMARIO

VOL. 1 N. 1 Dicembre 1978 - Febbraio 1979

In copertina:
Dall'età della pietra al Personal Computer.
Grafica di Renato Nisi

INDICE INSERZIONISTI

DIRETTORE RESPONSABILE:
Giampietro Zanga

GRAFICA E IMPAGINAZIONE:
Job Line srl

CONSULENZA E COORDINAMENTO:
Marcello Montedoro

DIFFUSIONE E ABBONAMENTI:
Gabriella Napoli, Silvia Decari

DIREZIONE, REDAZIONE:
P.le Massari, 22 - 20125 Milano
Telefoni 68.03.68 - 68.00.54

AMMINISTRAZIONE:
Via Vincenzo Monti, 15 - 20123 Milano

PUBBLICITA': Concessionario per l'Italia e l'Estero Reina & C. S.n.c. - P.le Massari, 22/A - 20125 Milano
Telefoni 60.35.15 - 69.04.91

Numero unico in attesa di autorizzazione del Tribunale di Milano

FOTOCOMPOSIZIONE: Tip. Coop. Parole Nuove - Brugherio

STAMPA: Litografia del Sole srl Buccinasco

Concessionario esclusivo per la diffusione in Italia e all'Estero:
SODIP - Via Zuretti, 25 - 20125 Milano
Via Serpieri, 11/5 - 00197 Roma

Spedizione in abbonamento Postale Gruppo IV/70

Prezzo della rivista L. 1.500

Numero arretrato L. 2.500
Abbonamento annuo L. 6.000
per l'Estero L. 9.000

I versamenti vanno indirizzati a: Jackson Italiana Editrice S.r.l.
Via V. Monti, 15 - 20123 Milano
mediante emissione di assegno bancario, cartolina vaglia o utilizzando il c/c Postale numero 11666203
Per i cambi d'indirizzo, indicare, oltre naturalmente al nuovo, anche l'indirizzo precedente, ed allegare alla comunicazione l'importo di L. 500, anche in francobolli.

EDITORIALE

Le colpe e le scelte

Parte della colpa è della fantascienza: una tranquilla massaia va al cinema a vedere «2001 odissea nello spazio», non capisce il film come non l'ha mai capito nessuno senza aver letto il libro ma rimane affascinata da HAL 9000; tornata a casa comincia a tempestare il figlio universitario che studia elettronica, rinfacciandogli i soldi spesi per farlo studiare, perchè vuole il computer che risponda al telefono, sbrighi le faccende di casa, faccia la dichiarazione dei redditi, si ricordi i compleanni eccetera eccetera. Ovviamente il figlio le risponde che comincerà a pensarci quando lei la smetterà di brontolare tutte le volte che lui si porta a casa una ragazza, ma questa è un'altra storia. Altrove ed in altri tempi un progettista di una grande ditta si legge «Nove volte sette», un gustosissimo raccontino di Asimov, scritto qualcosa come sette anni fa, secondo cui in un futuro che allora sembrava remoto un hobbista scopre che i conti si possono fare anche a mano e che vengono bene quanto quelli fatti con la calcolatrice tascabile (alla fine si uccide per via del rimorso, perchè il governo comincia ad installare uomini, che costano molto meno, al posto dei calcolatori nei missili); a rileggerlo oggi il racconto suona già antico (la calcolatrice tascabile «ronza per qualche secondo prima di fornire il risultato della moltiplicazione»), ma ormai il danno è fatto: molti oggi si arrangiano ancora con le somme, ma quasi tutti, senza la macchinetta in tasca, si perdono di fronte ad una moltiplicazione.

Un'altra parte della colpa va attribuita alle Pubbliche Amministrazioni dei diversi Paesi, che provano un ineffabile piacere nel rendere la vita dei cittadini quanto più complicata possibile: di fronte ad un moderno modulo delle tasse si incominciano a rimpiangere i tempi in cui gli sgherri del signorotto locale calavano come cavallette, si prendevano quello che volevano (non esclusa la moglie del padrone di casa) e se ne andavano senza seminare ulteriori angosce e terrori; oggi le pene per un conto sbagliato o per una fattura dimenticata in un cassetto tendono velocemente ai duecento anni di galera ed ai tre miliardi di multa, e le relative angosce affliggono tutte le notti del tapino che queste cose le prende sul serio.

Di fronte a simili raffinatezze della vita moderna è irresistibile la tentazione di prendere in blocco tutti questi terrori e queste preoccupazioni, farne un grosso pacco e scaricarlo ad un computer, il quale sarà magari anche lo scemo ad alta velocità della definizione classica, ma da pignolo qual è può ben tener testa a qualsiasi burocrate: l'idea di poter prendere ingiunzioni, tratte, solleciti, errori contabili e simili, di infilarli nell'apposito computer che sta vicino al telefono di casa e di trarne in cambio istruzioni chiare e cifre incontrovertibili è tale da solleticare chiunque.

Con queste premesse l'evoluzione dei computer ha preso una strada inevitabile ed inesorabile, il cui punto di arrivo si intravvede già con molta chiarezza: una macchina per ogni casa, a basso costo, facile da programmare e da interpretare, che si occupi di tutte quelle faccende che sono troppo stupide, troppo meccaniche o troppo ripetitive per occupare il prezioso tempo di un essere umano.

Molta acqua ha ancora da passare sotto i ponti, prima che questo obiettivo possa essere raggiunto: ancora per un po' la massaia dovrà rassegnarsi a battere i tappeti ed a fare ordine con le sue mani. Ma oggi l'acqua fa presto a passare. Quando abbiamo letto per la prima volta racconti di fantascienza come quello citato sopra, confusamente molti di noi hanno pensato «per fortuna quando e se questo si verificherà io certo non ci sarò più»; ebbene, eccoci qua, vispi, arzilli e ancora quasi giovani, ad accorgerci che la fantascienza ha sbagliato per difetto.

Due scelte, quindi: o guardare impotenti i mostri computerizzati infilarsi in ogni casa, scuola ed ufficio, chiedendoci dove andremo a finire; o prendere parte attiva al cambiamento, rendendoci bene conto che il mostro, se mai, è l'uomo, e che il computer non è più mostruoso di un cacciavite (che, come è noto, può essere usato per aggiustare una macchina come per piantarlo nel cuore di un altro essere umano per futili motivi). Noi alla Jackson la scelta l'abbiamo fatta, ed è questa rivista, BIT. Per ora uscirà trimestralmente e si farà in quattro per informarvi e tenervi aggiornati in tutti quei campi in cui fra l'uomo ed il computer non c'è di mezzo un'azienda: hobbies, gestione dei conti privati, piccole macchinette utili e quasi intelligenti per uso personale. E come tutto questo cambia, noi ve lo diremo prima.

Gianni Lange

SINTESI

Dal microprocessore all'informatica di massa

Il discorso microprocessori necessita di ulteriori puntualizzazioni in relazione al consolidamento delle applicazioni di questo componente nelle abituali aree di utilizzo, e di conseguenza in relazione al maggior grado di conoscenza riscontrabile in tutti quei settori che hanno tratto vantaggio da questa innovazione tecnologica.

Innanzitutto il discorso sul componente oggi si pone in maniera diversa da quando, circa sei anni fa, comparvero i primi microprocessori, quanto meno perchè, una volta acquisito il fatto tecnologico puro e semplice, si è compreso il legame con tutta una serie di altri fattori su cui ormai è possibile fare significative valutazioni. Pertanto la battaglia tra i microprocessori a 16 bit, che già si annuncia, dovrebbe avere un impatto con una realtà ben più matura e preparata di quella che, invece, incontrarono i primi microprocessori.

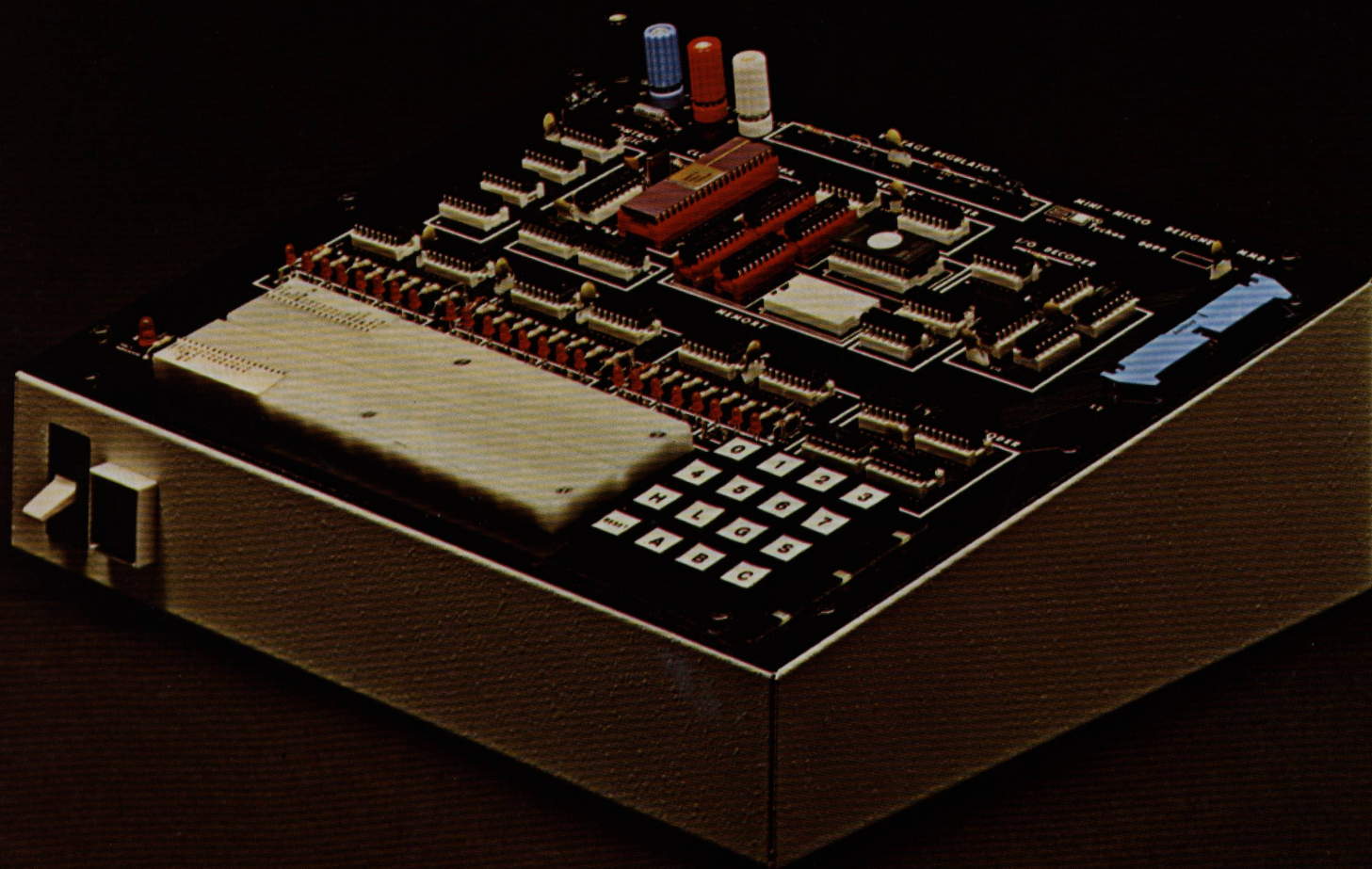
Ma è altrettanto importante avere acquisito in questo tempo la dimensione di una problematica che va al di là del componente. Con riferimento al fatto progettuale, già emergono le punte di una impostazione che, abbandonando l'iniziale accostamento al microprocessore come immediata ed economica alternativa alla logica cablata, ne coglie le possibilità progettuali più ampie e quindi più economiche.

Senz'altro sulla lentezza di questo processo ha pesato una evoluzione tecnologica molto più rapida dell'acquisizione concettuale; se poi si considera che tale acquisizione è stata, in un modo o nell'altro, pilotata dalle stesse case produttrici di componenti, ci si rende conto del perchè si è avuta una accentuazione del fattore componente rispetto a quello di sistema, e del fattore hardware su quello software.

Gli attuali costi hardware, decisamente bassi, la notevole potenza di calcolo delle varie unità di elaborazione, di qualunque tipo esse siano, il costo software non indifferente, ormai orientano la progettazione verso una direzione in cui i termini di multiprocessing e di sistema operativo sono i termini più attuali.

Queste considerazioni, alla cui base c'è la constatazione di una maggiore maturità del settore, già da sole evidenziano la necessità di una rivista esclusivamente dedicata ai microprocessori.

D'altra parte è sembrato opportuno considerare, per la stretta connessione che esiste di fatto, quell'area di applicazioni del microprocessore che in termini generali è possibile definire area del personal computer, a cui fanno o potranno fare capo esigenze di vario tipo, e comunque portatrice di istanze di linguaggi di programmazione «user-oriented», di periferiche a basso costo, di compatibilità di sistemi che concretizzano, fuori da ogni previsione, una tendenza ad una «informatica di massa» i cui sviluppi meritano senz'altro di essere seguiti.

Da questa premessa appare evidente quale sarà la fisionomia di BIT, rivista di microprocessori, rivista di hardware, rivista di software.

NEWSLETTER

Con LX 500 la Logabax si rivolge al piccolo utente

Con la spesa orientativamente valutabile in 3,5 milioni un imprenditore francese di modeste dimensioni può acquistare un sistema caratterizzato da un microprocessore Z80, 1,5 Kbytes di ROM, 16 Kbytes di RAM, un minidisco da 90 Kbytes, due unità I/O, un terminale portatile dotato di tastiera e capacità di stampa. È questa la conformazione operativa minima del nuovo minicomputer individuale annunciato dalla Logabax in occasione dell'ultimo SICOB. Il calcolatore si chiama LX500 e la sua commercializzazione oltre che dai concessionari Logabax verrà curata in Francia da società di servizi e da alcune organizzazioni di vendite al dettaglio come la già contattata Euro-Computer Shop di Parigi. La Logabax è una azienda leader sul mercato della «périinformatique» con all'attivo una vasta esperienza in atto di fabbisogni delle piccole e medie aziende. Di uno dei suoi sistemi più noti, LX 200, ne sono stati venduti 85.000 esemplari di cui 50 mila all'estero.

La Apple Computer ai primi posti

Mr. Steven Jobs e Mr. Stephen Wozniak (23 e 27 anni rispettivamente), due Steves come amichevolmente sono chiamati, possono ben essere contenti. La società da loro fondata nel 1976 con l'aiuto finanziario della Venroch (gruppo Rockefeller), la Apple Computer, nell'esercizio al 30 settembre scorso ha venduto calcolatori personali per 10 milioni di $, cifra che la pone ai primissimi posti del settore. Introdotto nello scorso anno l'Apple II è il suo attuale cavallo di battaglia. Nella versione base costa un migliaio di dollari, il triplo se dotato di stampante, video, interfacce telefoniche e memoria aggiuntiva.

Parità fra piccoli e grossi computer

Alla fine del 1979 tutte le aziende con un giro di affari compreso fra 200 mila dollari e 4 milioni di $ potranno acquistare un piccolo sistema gestionale con meno di 10.000 dollari, ivi compreso il software di base necessario al loro funzionamento.
Attualmente per un sistema delle stesse capacità occorre una cifra del 50% superiore. È questa l'opinione espressa da Ryal Poppa, numero uno della Pertec Computer (un gruppo in forte crescita) intervenendo ad un simposio sulla tecnologia. A suo avviso nel giro di alcuni anni le vendite di personal computers potrebbero eguagliare il livello di fornitura delle unità più grosse.

1.000 PET per l'Europa

Il PET, il minicalcolatore di basso costo (circa un milione di lire) della Commodore International viene ora assemblato anche in Europa (Gran Bretagna) ad un ritmo di circa 500 unità al mese.
Già però il costruttore ha disposto di raddoppiare la produzione sì da portarla assai vicino ad un terzo di quella americana, oggi pari a 4.500 pezzi al mese.
Il 60% dei PET venduti trova utilizzazione nel settore educativo e dell'insegnamento. Anche se la produzione di questo personal computer è stata sino a ieri fatta negli USA circa i due terzi di essa hanno trovato sistemazione al di fuori degli Stati Uniti.

Quanto rende vendere personal computer

A spulciare il «Retail Computer Buyers Survey», edizione 78, si ricava ad ogni paragrafo l'impressione che la domanda di calcolatori personali superi l'offerta di un margine talmente ampio da giustificare l'entrata in scena e il successo di fabbricanti anche non proprio ortodossi, commercialmente discorrendo. Estrema facilità quindi ad entrare sul mercato del personal-computing? Si risponde, previo alcune considerazioni, l'indagine della Image Resource per chi è in possesso di talento tecnico e di adeguati mezzi finanziari. Le possibilità di guadagno sono proporzionate al ricorrere e al combinarsi di queste due condizioni. In genere, ha calcolato la Image Resource, il personal computer frutta ad un piccolo rivenditore margini di utili dal 25 al 30% sul prezzo di vendita. Quest'anno, è sempre la stessa fonte ad informare, tale attività dovrebbe generare affari per 200 milioni di $ di cui 120 realizzati tramite stores. L'importanza di questi ultimi, come canale di distribuzione, è destinata a crescere se come prevede l'indagine della Image Resource nel 1980 su un fatturato previsto di 425 milioni, un fetta vicino ai 300 milioni toccherà agli stores.

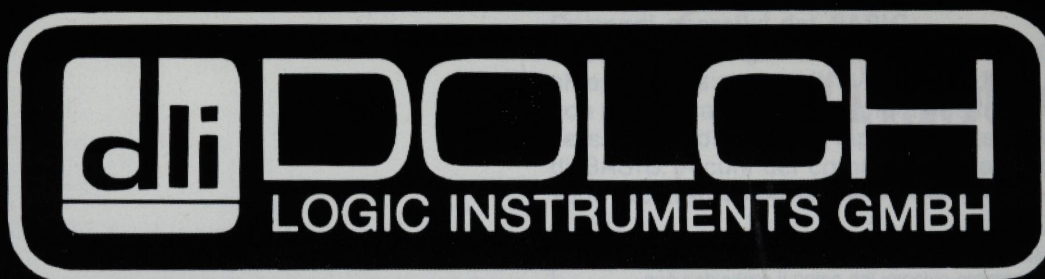
Sistemi DEC acquistabili al banco

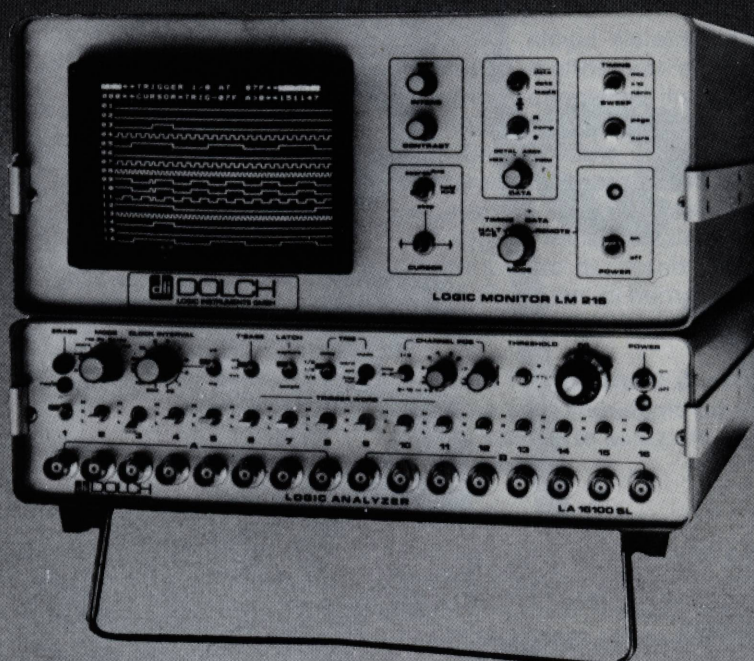
Dei grossi produttori di computer, mini o maxi, la Digital Equipment è stata la prima a armarsi di coraggio ed entrare di persona sul mercato al dettaglio di prodotti elettronici. Lo ha fatto aprendo, a Boston, uno store che nelle sue intenzioni dovrebbe fungere da esperimento e da assaggio circa l'atteggiamento da assumere nei confronti di una tendenza non più ristretta a pochi amatori-intenditori ma che va interessando una schiera molto vasta di utenti compresi commercianti, professionisti e piccoli imprenditori. È proprio a questi che l'iniziativa della DEC si rivolge proponendo sistemi dai costi variabili da 10 a 20 mila dollari alla cui assistenza provvede la stessa Digital attraverso gli uomini del Commercial OEM group. Un secondo negozio si sta programmando di aprirlo nella stessa area di Boston, destinando anch'esso alla vendita di una gamma mediana e sufficientemente estesa di strumenti elettronici da calcolo, forse con qualche concessione in più al piccolo utente.

Anche la ITT entra in scena

Mesi fa la ITT aveva siglato con la Apple Computer un accordo di collaborazione. Poche settimane sono bastate per mettere a punto un minicalcolatore personale imperneato su un microprocessore 6502 della MOS Technology. La macchina è stata presentata all'ultimo SICOB durante il quale si è pure appreso che essa sarà commercializzata dalla rete consumer. (Oceanic per la Francia) e non da quella EDP (ITT Data Systems). Il calcolatore si compone di una tastiera, un registratore, un display e, come option, di una stampante da 24 linee di 40 caratteri.

Standards necessitano

Il consumo di personal-computer cresce ma il mercato manca di orientamenti quanto a strategie produttive e commerciali.
Perchè tanto interesse verso questo prodotto? Alcuni fabbricanti collegano tale interesse alla simpatia mostrata dal pubblico verso i videogames e di conseguenza offrono «calcolatorini» a basso costo e di limitate prestazioni, accessibili però ad un'utenza molto vasta. Altri costruttori, invece pensano che il successo sia da giustificare con argomentazioni più serie ritenendo il fenomeno una naturale espansione verso il basso del data processing. Alla luce di quanto si è potuto osservare fino ad ora pare siano questi (Commodore, Pertec, etc.) ad aver visto meglio.
Quanto al supporto hardware esiste ancora divergenza fra televisore e display dedicato. Analoga situazione di incertezza sussiste a livello di software: proporre programmi fissi o soluzioni programmabili!

ANALIZZATORI DI STATI LOGICI

Con pilotaggio a microprocessori, grande formato di memoria e con 8, 16 oppure 32 canali. Costruito in Germania.

- 8, 16 o 32 canali
- Fino a 2048 Bit/canale
- Memoria di riferimento interna
- Accesso digitale alla memoria di riferimento
- Soglia logica regolabile
- Presentazione dati in esadecimale, ottale ASCII Mnemonica
- Trigger combinatorio
- Acquisizione dati in modo sincrono od asincrono
- Programmazione esterna
- Ricerca automatica dei guasti
- Presentazione automatica delle condizioni di errore
- Memoria a display organizzata da microprocessore

TECNICHE ELETTRONICHE AVANZATE S.a.s.

20147 MILANO - VIA S. ANATALONE, 15 - TEL. 415.87.46/7/8
00187 ROMA - VIA DI P.TA PINCIANA, 4 - TEL. 480.029-465.630
INDIRIZZO TELEGRAFICO: TELAV - MILANO - TELEX: 39202

NUCLEO

Il BASIC (Beginners All-purpose Symbolic Instruction Code) è uno dei più diffusi e semplici linguaggi di programmazione ad alto livello. È, ormai, il linguaggio più utilizzato sui personal computers. In questo articolo sono descritte le istruzioni comuni alle varie versioni, con l'ausilio di significativi esempi.

Introduzione al BASIC

di Robert Baker

Il BASIC, sviluppato nei primi anni '60 all'Università di Dartmouth, è un linguaggio di programmazione di tipo conversazionale.

Il BASIC è applicato sia in campo scientifico e commerciale che in campo educativo, dal momento che può essere usato per risolvere problemi matematici semplici e complessi. I comandi e le istruzioni di programmazione sono di facile comprensione e di numero ridotto, per cui il BASIC risulta essere uno dei linguaggi più semplici da imparare e da usare.

Come molti linguaggi di programmazione, il BASIC può essere diviso in due sezioni: 1) istruzioni elementari per programmi semplici, 2) istruzioni e tecniche più progredite per programmi complessi.

Questo articolo di introduzione al Basic, si occupa, fondamentalmente, delle funzioni e delle istruzioni elementari ammesse dalle versioni più comuni di BASIC e ne illustra le applicazioni con alcuni esempi. Si accennerà inoltre a un certo numero di comandi operativi, e a qualcuna delle prestazioni più avanzate riscontrate in alcuni sistemi che utilizzano il BASIC.

Le prestazioni specifiche di ogni computer presentano differenze minime fra loro per cui si hanno molti differenti dialetti del linguaggio BASIC. Quando si scrivono programmi che devono essere compatibili con sistemi diversi, si usano di norma soltanto istruzioni elementari che sono comuni a molte forme di BASIC. Si dovrebbe, pertanto, sempre consultare il manuale di programmazione BASIC (o una documentazione equivalente) per verificare i comandi e le istruzioni in realtà disponibili e le convenzioni usate per un particolare sistema.

Comandi operativi del BASIC

Dopo aver acquisito l'accesso al BASIC nel modo richiesto dal sistema, il BASIC risponde normalmente con un messaggio che segnala attesa di comando. I diversi comandi operativi dipendono dalla versione usata di BASIC. Alcuni dei comandi abitualmente disponibili sono illustrati in fig. 1.

OLD: carica un programma precedentemente salvato. Il BASIC può richiedere il nome del vecchio programma (o flusso), a seconda del sistema.
NEW: comanda l'input da console di un nuovo programma. Il BASIC può richiedere un nome di programma (o flusso) a seconda del sistema.
LIST: stampa il programma corrente sulla console. Alcuni sistemi permettono di stampare o linee singole o gruppi di linee.
RUN: compila ed esegue il programma corrente. Alcuni sistemi permettono che si indichi la linea di inizio esecuzione.
SAVE, RESAVE, o REPLACE:
salva il programma corrente.
UNSAVE, PURGE, o SCRATCH:
cancella il programma corrente.
BYE, GOODBYE, o SYSTEM:
il BASIC è scaricato.

Figura 1 — Comandi operativi del BASIC.

Inoltre, alcuni sistemi possono accettare comandi abbreviati, o offrire vari comandi di «editing», con i quali è possibile cancellare linee, numerare in sequenza le linee di programma, ecc. La maggior parte degli interpreti BASIC usano scrivere READY quando aspettano un comando operativo. Si inseriscono nuove linee di programma scrivendo semplicemente un numero di linea seguito dalla istruzione desiderata, e battendo per ultimo il tasto di ritorno carrello. Saranno trattate più avanti le nozioni di numeri di linea e di istruzioni individuali.

Concetti fondamentali di programmazione

Come ogni linguaggio ad alto livello per computer, il BASIC usa certe notazioni per i numeri, le variabili, e le operazioni.

Numeri

Il computer può calcolare il valore di espressioni come 12/7 o $\sqrt{9}$ ma, pur usando numeri, queste espressioni non sono considerate numeriche, e perciò non possono essere usate in liste di dati costanti.

I numeri cioè sono valori autodefiniti e sono generalmente espressi in forma decimale. Inoltre possono essere positivi o negativi (Es.: 3 o - 2,79).

La maggior parte dei linguaggi BASIC permette altresì una notazione scientifica per i numeri, usando la lettera E per indicare 10 elevato a una potenza. Ad esempio un numero espresso nella forma

aEb vuol dire $a \times 10^b$

Allora per esprimere 0,125 si può scrivere:
0,125E0 oppure 125E-3.

In maniera analoga 1000 si può scrivere come
1E3 oppure 10E2.

Con la notazione scientifica, deve sempre comparire un numero su entrambi i lati della lettera E; questo numero può essere positivo o negativo. Esiste abitualmente un numero massimo di cifre permesso per le due notazioni numeriche, in base al tipo di computer e alla versione di BASIC.

Variabili

Le variabili numeriche semplici sono generalmente espresse da una singola lettera o da una lettera seguita

da una singola cifra. Esempi di variabili sono:
E7, A, B, C2, X0

Alcuni computers consentono altre definizioni di variabile, ma se si vuole compatibilità con altri sistemi BASIC, è preferibile utilizzare questa forma ridotta e semplice.

Le variabili numeriche rappresentano un valore numerico. Il valore è assegnato alla variabile mediante le due istruzioni LET o READ.

Un valore assegnato può cambiare solo con un'altra istruzione LET o READ, che cambia esplicitamente la variabile.

Nel tipico interprete BASIC, tutte le variabili numeriche sono inizialmente poste uguali a 0; è necessario perciò assegnare un valore iniziale alla variabile laddove si desidera un valore diverso da 0.

Operatori aritmetici

I calcoli aritmetici, in un linguaggio come il BASIC, si effettuano valutando le espressioni di ogni linea di programma, che si presentano in una forma analoga a quella usata nei calcoli matematici standard.

Tutti gli operatori aritmetici elencati nella Tabella 1 possono essere usati per scrivere l'espressione desiderata. In calcoli che utilizzino più operatori, l'ordine di precedenza è determinato da una serie convenzionale di regole:

1. Una espressione che compare in parentesi è valutata prima che la intera quantità messa fra parentesi sia usata nel calcolo.
2. Normalmente, un operatore non può seguire un altro operatore. Tuttavia gli operatori + e—possono seguire gli operatori *, /, ↑ o **. In questo caso, il + o il – indica il segno della quantità su cui si interviene con l'operatore principale.
3. In una espressione senza parentesi, **e ↑ hanno la precedenza su *e /, che a loro volta hanno la precedenza su + e –.
4. Se compaiono soltanto gli operatori *e /, in assenza di parentesi, il BASIC fa le operazioni da sinistra a destra cioè nell'ordine di lettura.
5. Se compaiono soltanto gli operatori +e –, in assenza di parentesi, il BASIC fa le operazioni da sinistra a destra cioè nell'ordine di lettura.

Gli esempi di fig. 2 aiuteranno ad illustrare queste regole e a chiarire il modo in cui l'interprete BASIC esegue tipicamente i vari calcoli.

OPERATORE	ESEMPIO	DESCRIZIONE
+	A+B	Si somma A a B
+	+A	A è positivo
-	A-B	Si sottrae B da A
-	-A	A è negativo
*	A*B	Si moltiplica A per B
/	A/B	Si divide A per B
↑	A ↑ B	A elevato alla potenza B
**	A** B	

Tabella 1–Operatori aritmetici.

```
B+3*A**2       ←→ A è elevato al quadrato, poi è moltiplicato per 3, e
                  infine sommato a B
B+(3*A)**2     ←→ Il prodotto di A per 3 è elevato al quadrato, poi som-
                  mato a B
(B+(3*A))**2   ←→ Il prodotto di A per 3 è sommato a B, poi la somma è
                  elevata al quadrato
A/B/C/         ←→ Il quoziente di A diviso B, è diviso per C
A-B-C          ←→ C è sottratto al risultato di A-B
A**2**3        ←→ Il risultato di A elevato al quadrato è poi elevato al
                  cubo
```

Figura 2 – Esempi di calcoli aritmetici.
Un tipico interprete di BASIC comprende un algoritmo di analisi delle espressioni aritmetiche che riconosce le usuali operazioni di addizione, sottrazione, moltiplicazione, divisione ed esponenziali. Nel testo è descritta la priorità, e si sono usate le parentesi per esplicitare l'ordine delle operazioni.

Funzioni aritmetiche

Oltre alle cinque operazioni aritmetiche, la versione completa di interprete BASIC è in grado di valutare diverse funzioni matematiche standard.

Nella Tabella 2 sono elencate le funzioni elementari disponibili in un tipico BASIC. Qualsiasi espressione purchè valida può essere sostituita ad X in tutte le funzioni. Spesso sono disponibili funzioni più complesse che saranno trattate in seguito.

Si possono avere anche altre funzioni matematiche a seconda del particolare computer usato, così come funzioni non aritmetiche.

Simboli relazionali

Quando è necessario confrontare dei valori, si usano sei simboli matematici di relazione abbinati alla istruzione IF.

Alcuni sistemi possono usare espressioni scritte sostitutive, specialmente quando i terminali non hanno i simboli standard.

La Tabella 3 elenca i simboli standard e le forme sostitutive di operazioni relazionali.

FUNZIONE			DESCRIZIONE
SIN(X)	Seno di X		Nella maggior parte dei sistemi X è abitualmente interpretato come un angolo espresso in radianti.
COS(X)	Coseno di X		
TAN(X)	Tangente di X		
COT(X)	Cotangente di X		
ATN(X)	Arcotangente di X		
EXP(X)	e elevato alla potenza X	(e^X)	X è sempre interpretato come numero.
LOG(X)	Logaritmo naturale di X	(ln X)	
SQR(X)	Radice quadrata di X	($\sqrt{X}$)	
ABS(X)	Valore assoluto di X	($\|X\|$)	

Tabella 2 – Tipiche funzioni matematiche del BASIC. L'implementazione usuale di un BASIC completo prevede questo elenco di funzioni matematiche residente in esso. La precisione e la grandezza dei valori numerici dipendono dai particolari della implementazione e dalla rappresentazione interna dei numeri.

SIMBOLO STANDARD	ESPRESSIONE SOSTITUTIVA	ESEMPIO	DESCRIZIONE
=	=	A=B	A è uguale aB
<	LT	A<B A LT B	A è minore di B
<=	LE	A<=B A LE B	A è minore o uguale a B
>	GT	A>B A GT B	A è maggiore di B
>=	GE	A>=B A GE B	A è maggiore o uguale a B
<>	NE	A<>B A NE B	A non è uguale a B

Tabella 3 — Tipici operatori relazionali del BASIC. Questi operatori si usano per eseguire confronti fra operandi che possono essere numeri, variabili o espressioni. In alcune implementazioni del BASIC, quando non si dispone di caratteri speciali, si ricorre a sostituti.

Identificatori di linea (numeri di linea)

In ogni programma BASIC, ciascuna istruzione è associata ad un numero di linea, come si è detto in precedenza.

La prima linea di solito ha il numero 10 per cui le linee seguenti sono generalmente numerate di 10 in 10. Questo permette di inserire istruzioni extra fra linee già esistenti, identificando la nuova istruzione con un numero di linea compreso fra le due linee adiacenti. Il BASIC inserisce correttamente le nuove linee usando il numero di linea che identifica l'istruzione. L'unica limitazione sui numeri di linea in un programma è che ogni numero di linea deve essere maggiore del precedente.

Istruzioni elementari di programmazione

Le istruzioni elementari del linguaggio BASIC si riconoscono dalla prima parola dell'istruzione stessa.

Istruzione LET

L'istruzione LET è usata per assegnare un valore numerico ben preciso (che può essere anche il risultato di un calcolo) ad una particolare variabile o ad un gruppo di variabili.

La forma generale della istruzione LET è:

LET [variabile] = [espressione][1]

In alcune versione BASIC, la parola LET è facoltativa nell'istruzione.

Esempi di assegnamento di una particolare variabile sono:

```
10    LET X=1
20    LET A=B*2/3
```

In diversi BASIC, si può assegnare lo stesso valore a più di una variabile con una sola istruzione LET.

Es.:

```
30    LET X=Y2=B=4
```

Inoltre, una variabile può comparire su entrambi i lati del segno uguale, nel qual caso l'operazione è fatta usando i valori «vecchi» dei dati, e quindi il risultato è assegnato alle variabili a sinistra del segno "=".

Ad esempio, l'istruzione

```
100    LET X=X+1
```

somma 1 al vecchio valore di X e considera il risultato come nuovo valore da assegnare a X.

L'istruzione LET non è una uguaglianza algebrica, ma un comando ad eseguire i calcoli e ad assegnare il risultato alle variabili specificate.

È un aspetto negativo dei linguaggi per computers il fatto che molti linguaggi usano lo stesso simbolo «=» sia per operazioni di assegnamento (trasferimento di dati), che per tests di relazione (confronto di dati).

Istruzioni READ e DATA

Queste istruzioni sono usate per far acquisire al programma che è in esecuzione su di un computer le informazioni necessarie alla elaborazione.

L'istruzione READ assegna i valori acquisiti tramite le istruzioni DATA, alle variabili elencate nella medesima istruzione READ. L'istruzione READ deve essere sempre abbinata alla istruzione DATA.

Prima che il programma venga messo in esecuzione, il tipico interprete BASIC prende tutte le istruzioni DATA nell'ordine in cui si presentano, e le memorizza in una zona dati. In seguito tutte le volte che incontra una istruzione READ, assegna alle variabili presenti nella istruzione READ i valori numerici nell'ordine con cui sono disposti nella zona dati. Se a fronte di una ulteriore richiesta di dati, questi risultano esauriti, il programma si ferma e viene stampato un messaggio di errore.

Le istruzioni READ sono normalmente poste all'inizio del programma, in quanto i dati devono essere forniti in tempo perchè il programma li possa elaborare. Le istruzioni DATA possono essere situate in qualsiasi punto del programma, purchè si presentino nell'ordine corretto. Molti programmatori trovano conveniente porre le istruzioni DATA prima della istruzione END, ossia alla fine del programma.

La istruzione READ ha la forma:

READ [elenco di variabili]

L'istruzione DATA ha la forma

DATA [elenco di numeri]

Esempio:

```
100    READ X,Y,Z
200    DATA 1,2,3,4,5,6,7,8,9,0
```

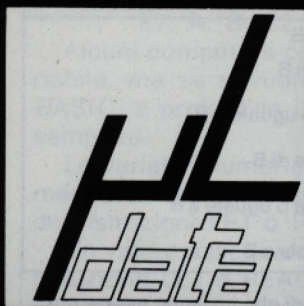
In questo esempio le istruzioni leggeranno i primi tre numeri del blocco dati (1, 2 e 3) e li assegneranno a X, Y e Z.

Va tenuto presente che solo numeri e non espressioni possono essere usati nelle istruzioni DATA, e che 14/3 e$\sqrt{7}$ sono espressioni!

Istruzione PRINT

Le istruzioni PRINT sono usate con svariate finalità. Sebbene il comando sia chiamato «PRINT», in molti sistemi significa in realtà «invio di dati verso il dispositivo di output primario del BASIC», quale può essere una stampante hard copy o un display elettronico. L'esempio che segue illustra il formato usato per stampare i risultati di un calcolo o un valore di variabile:

```
10 PRINT X,SQR(X)
```

(1) Ciascuna voce che compare in parentesi indica una forma generale di un dato argomento.
Esempio:
[variabile]= una qualunque variabile
[numero di linea]= un qualunque numero di linea

MICROLEMdata

Sistemi per l'informatica

STEVENS INCORPORATED ARNOLD

- Alimentatori non interrompibili con batterie di back-up
- Convertitori DC/DC
- Chopper DC/AC
- Trasformatori ad alto isolamento

NATIONAL MULTIPLEX CORP

- Sistemi di sviluppo universali per microprocessori
- Mini sistemi di gestione
- Digital Cassette Recorders

...e non è tutto!

Venite al

XV BIAS

stand M 38-40
Pad. 13

Honeywell

Honeywell Information Systems Italia

Stampanti seriali con velocità di stampa da 30 a 160 CPS, bidirezionali con ottimizzazione del percorso.

Questa istruzione farà stampare (o visualizzerà su display) il valore corrente di X e la radice quadrata di X sulla medesima riga.

Per stampare uno specifico messaggio, si racchiude il messaggio fra apici e si usa il seguente formato:

20 PRINT "NESSUNA RISPOSTA"

Questa istruzione farà stampare le parole NESSUNA RISPOSTA su una singola linea e senza gli apici.

Volendo, si possono unire messaggi e valori:

30 PRINT "LA RADICE QUADRATA DI" X "È" SQR(X)

Questa istruzione permette di scrivere la riga seguente (se il valore di X è 9):

LA RADICE QUADRATA DI 9 È 3

Per saltare una linea o avere soltanto un ritorno di carrello e un avanzamento di linea, si usa l'istruzione PRINT senza argomento:

40 PRINT

In molti sistemi BASIC avanzati ci sono diversi caratteri di controllo della stampa, e speciali istruzioni che permettono di controllare la spaziatura dei dati quando questi vengono stampati. Tale controllo può variare da un sistema all'altro, ma questo esula dai limiti che questo articolo si propone. Per il momento è sufficiente capire l'uso principale della istruzione PRINT e conoscere i metodi comunemente utilizzati per stampare dati in un qualsivoglia formato desiderato.

Istruzione GO TO

L'istruzione GO TO è usata per saltare ad una data istruzione del programma, diversa da quella immediatamente successiva. La linea a cui si salta può trovarsi in qualsiasi punto del programma, prima o dopo la linea corrente.

Il formato della istruzione GO TO è:

GO TO [numero di linea]

Esempio:

100 GO TO 200

In questo esempio il programma salterà alla linea 200 ogni volta che raggiunge la linea 100.

Istruzione IF

L'istruzione IF è usata per saltare ad una data istruzione, ma solo se una certa relazione è verificata: in altre parole, il programma salterà al numero di linea designato solo se la relazione specificata è vera.

L'istruzione IF, che talvolta è designata come istruzione di GO TO condizionato, ha il seguente formato:

IF [espressione] [relazione] [espressione], THEN [numero della linea]

Nell'esempio di fig. 3, se il valore di X è 0, il programma salterà alla linea 200. Se il seno di X è minore o uguale a 0,5, il programma salterà indietro alla linea 80. Se entrambe le condizioni sono false, il programma andrà alla linea che segue la linea 130.

La parola «THEN» può essere sostituita da «GO TO» e la virgola che precede «THEN» può essere facoltativa, a seconda del tipo di sistema.

Figura 3 — Esempi di istruzioni IF.

Istruzione ON

Mentre l'istruzione IF fornisce al programma una biforcazione in due rami, l'istruzione ON (se è implementata) permette di scegliere una molteplicità di ramificazioni.

Il formato generale della istruzione ON è:

ON [espressione], GO TO [numero di linea, numero di linea,...]

Si può usare qualsiasi espressione e l'istruzione può contenere più di un numero di linea, purchè l'istruzione sia contenuta in una sola linea.

Il valore dell'espressione è calcolato e troncato ad un numero intero. Se il risultato è 1, il programma salta al primo numero di linea dell'elenco; se il risultato è 2, passa al secondo numero di linea dell'elenco, ecc. Se il risultato dell'espressione è minore di 1 o supera il numero degli identificatori di linea presenti nella istruzione, viene stampato un messaggio di errore.

Nell'esempio di figura 4 è possibile uno dei casi seguenti:

Se X=2, il programma andrà alla linea 100
Se X=3, il programma andrà alla linea 200
Se X=4, il programma andrà alla linea 300
Se X è minore di 2 o è maggiore di 4, verrà stampato un messagio di errore.

In alcuni sistemi si può omettere la virgola che precede «GO TO», o si può sostituire «GO TO» con la parola «THEN».

Istruzione END

Tutti i programmi devono avere necessariamente l'istruzione END, e questa deve essere sempre l'istruzione con il più alto numero di linea nel programma.

Figura 4 — Esempio di istruzione ON

Il formato generale della istruzione END è semplicemente la parola «END»:

9999 END

Istruzione STOP

L'istruzione STOP equivale all'istruzione GO TO, in cui la linea a cui si salta è l'istruzione END del programma.

L'istruzione consiste semplicemente nella parola «STOP»:

1000 STOP

Istruzione REM (commento)

L'istruzione REM fornisce un metodo per inserire osservazioni e commenti nel «listing» di un programma.

Le istruzioni GO TO, IF, ON, o GOSUB possono usare l'identificatore di linea relativo all'istruzione REM in quanto il contenuto della linea con REM è ignorato.

Il formato generale della istruzione REM è:

REM [commento]

Esempio:

1000 REM QUESTA ROUTINE CALCOLA LA MEDIA DI 10 NUMERI

Esempi elementari di programmazione

Ora che sono state esposte le istruzioni e le funzioni elementari del BASIC, è possibile illustrare con tre programmi come si combinano le istruzioni per eseguire svariati calcoli o confronti.

zione NEXT deve essere la stessa che compare nella istruzione FOR.

I loops possono essere l'uno dentro l'altro fino ad un certo numero dipendente dalla particolare versione BASIC, ma vanno inseriti seguendo una regola precisa (e cioè un loop non deve incrociare un altro loop).

La figura 5 mostra alcuni esempi di loops multipli, alcuni corretti ed altri errati.

Il valore della variabile numerica parte dal valore fissato in espressione$_1$ e viene modificato della quantità espressa in espressione$_3$ (positiva o negativa) ogni volta che il BASIC termina un ciclo di loop. Se sono omesse la parola «STEP» e la quantità di espressione$_3$, il valore di STEP è considerato per difetto uguale a +1.

Va notato tuttavia che, volendo, la variabile di controllo può essere cambiata all'interno del loop di programma.

Bisogna stare attenti quando si utilizzano valori relativi a STEP non interi, dal momento che i numeri non interi non vengono rappresentati esattamente nel sistema binario, e il computer può troncarli o approssimarli. Questo può determinare l'esecuzione di un numero di loops non voluto. Pertanto bisognerebbe fare ogni sforzo per usare un valore relativo a STEP intero, tutte le volte che sia possibile.

Figura 5 — Loops leciti e loops errati. Esistono modi corretti e modi errati per inserire un loop dentro un altro loop. Il modo corretto è quando il programma vede sempre i loops interni interamente contenuti dentro i loops esterni. Il modo scorretto è quando si hanno dei loops che si incrociano reciprocamente.

LOOPS

Il modo più semplice per eseguire una sequenza ripetitiva di istruzioni è l'impiego di un loop di programma. Il blocco di istruzioni contenute nel loop viene eseguito ripetutamente, fino a che non risulta soddisfatta una data condizione.

Nel BASIC si usano due istruzioni per specificare un loop: l'istruzione FOR, che si usa all'inizio del loop, e l'istruzione NEXT, che si usa alla fine.

Il formato generale dell'istruzione FOR è:

FOR [variabile numerica]=[espressione$_1$] TO [espressione$_2$] STEP [espressione$_3$]

A seconda della versione di BASIC, nella istruzione FOR alla parola «STEP» si può sostituire la parola «BY».

L'istruzione NEXT ha il formato:

NEXT [variabile numerica]

In ciascun loop la variabile che compare nella istru-

Nell'esempio di figura 6, il loop di programma sarà eseguito 10 volte, perchè X è incrementato di 1 a partire da 1 e fino al valore 10. I numeri X e X-fattoriale saranno stampati durante ogni ciclo di loop.

```
 5 LET F=1
10 FOR X=1 TO 10 STEP 1
20 LET F=F*X
30 PRINT X,F
40 NEXT X
```

Figura 6 — Esempio di Loop.

In questo esempio l'istruzione FOR poteva essere abbreviata in:

10 FOR X=1 TO 10

dal momento che il valore relativo a STEP è 1.

Subroutines

Quando una particolare sequenza di istruzioni deve essere usata ripetutamente in molti punti diversi all'interno di un programma, queste possono essere scritte sotto forma di subroutine.

Una subroutine è un programma completamente autocontenuto a cui si accede dal programma principale o da altre subroutines usando la convenzione di «chiamata a subroutine».

Nel BASIC, le subroutines vengono chiamate usando, in qualsiasi punto del programma principale (o di altre subroutines), l'istruzione GOSUB, che ha il formato generale:

GOSUB [numero di linea]

L'istruzione GOSUB è simile alla istruzione GO TO, tranne che il computer salva la locazione dell'istruzione GOSUB prima di eseguire la subroutine vera e propria.

Le istruzioni della subroutine vengono eseguite di seguito, fino ad incontrare una istruzione RETURN.

L'istruzione RETURN ordina al computer di tornare alla linea che segue l'istruzione GOSUB che ha chiamato la subroutine stessa.

Il formato di questa istruzione è costituito semplicemente da un numero di linea e dalla parola «RETURN»:

1000 RETURN

ed è di solito l'ultima linea della subroutine.

Le subroutines possono essere scritte in un qualsiasi punto del programma principale, ma vengono eseguite esclusivamente con una istruzione GOSUB; analogamente si esce da una subroutine soltanto attraverso l'istruzione RETURN.

Una subroutine può contenere più di una istruzione RETURN, quante, cioè, sono richieste dal particolare problema che è trattato.

Una subroutine può chiamare un'altra subroutine usando una istruzione GOSUB dentro la subroutine «chiamante».

```
 10 LET L=4
 20 PRINT "QUESTO PROGRAMMA STAMPA"  "............"
 30 GOSUB 190
 40 PRINT "QUESTO MESSAGGIO CON"     "............"
 50 GOSUB 190
 60 PRINT "QUATTRO LINEE BIANCHE"    "............"
 70 GOSUB 190
 80 PRINT "TRA LE LINEE DI MESSAGGIO"
 90 GOSUB 190
100 REM poi sono stampati i numeri da 1 a 10
110 REM seguiti ciascuno da un numero di linee bianche
120 REM pari al numero stampato in testa.
130 FOR L=1 TO 10
140 PRINT L
150 GOSUB 190
160 NEXT L
170 PRINT "ESEGUITO"                 "............"
180 STOP
190 REM Subroutine per stampare «L» linee bianche
200 IF L>0, THEN 220
210 RETURN
220 FOR X=1 TO L
230 PRINT
240 NEXT X
250 RETURN
260 END.
```

Figura 7 — Esempi di Subroutines.

Il numero di inserimenti di subroutines dipende dalla particolare versione di BASIC che è usata.

L'esempio di fig. 7 illustra come sono utilizzate le subroutines, e l'uso di due istruzioni RETURN in una sola subroutine. Mostra inoltre come il numero di linea di una istruzione REM può essere usato in una istruzione GO TO, IF, ON, o GOSUB, come si è detto in precedenza.

In questo esempio ogni volta che si incontra l'istruzione «GOSUB 190», il programma andrà alla subroutine che parte alla linea 190. Se il valore di L è minore o uguale a 0, il computer tornerà al programma principale. In caso contrario, la subroutine stamperà un numero di linee bianche uguale al valore di L, prima di tornare al programma principale.

Elenchi e tabelle

Oltre al limitato numero di variabili semplici, di solito il BASIC offre la possibilità di definire gli elementi di elenchi o tabelle.

Le variabili semplici vengono perciò chiamate variabili senza indice, mentre gli elementi di un elenco o tabella si chiamano variabili con indice.

Le variabili con indice possono essere utilizzate in ogni punto in cui è usata una variabile senza indice, tranne che subito dopo «FOR» in una istruzione FOR o in una istruzione NEXT.

Tratteremo più avanti alcune delle istruzioni e funzioni speciali che operano su matrice.

Il nome di ciascun elenco o tabella deve essere designato con una lettera. Ciascun elemento di un elenco (detto anche vettore) è individuato con il nome dell'elenco seguito da un indice fra parentesi:

A(0), A(1),...A(n)

Gli elementi di una tabella (detta anche matrice) sono individuati con due indici:

B(1,1), B(1,2),...B(m,n)

Il formato dell'indice è flessibile; può essere sia una costante, sia una qualunque espressione lecita, a patto che il valore dell'indice non sia minore di 0.

La lettera che denota il nome di elenco o di tabella può essere usata senza pericolo di confusione anche come variabile semplice; ma non si può usare la medesima lettera per denotare sia un elenco che una tabella nello stesso programma, poichè un elenco è di fatto una tabella costituita da una sola colonna.

La routine contenuta nell'esempio di fig. 8 mostra come si può costruire in un programma una tabella di dati 3 x 4 usando istruzioni elementari, e come i vari elementi della tabella sono, a questo punto, individuati.

```
10 FOR I=0 TO 2
20 FOR J=0 TO 3
30 READ M(I,J)
40 NEXT J
50 NEXT I
60 DATA 1,2,3,4
70 DATA 5,6,7,8
80 DATA 9,10,11,12
```

Figura 8 — Esempio di costruzione di una Tabella-dati.

L'area di memoria necessaria ad ogni elenco o tabel-

la è riservata automaticamente fintanto che gli elementi sono minori di 10 (il numero esatto dipende dalla versione di BASIC). Per indici più grandi, l'area è riservata usando l'istruzione DIM.

L'istruzione DIM può comparire in qualsiasi linea prima della istruzione END, ma normalmente è posta all'inizio del programma. L'istruzione DIM può essere usata anche per riservare l'area di memoria necessaria ad un elenco o tabella di minime dimensioni quando la presenza di un programma esteso tende ad occupare più area di memoria.

Il formato generale della istruzione DIM per elenchi o tabelle è:

DIM |nome dell'elenco| (indice massimo)
DIM |nome della tabella| (indice massimo di riga, indice massimo di colonna)

Non sono necessarie istruzioni DIM distinte per ogni elenco o tabella; si possono specificare diversi elenchi o tabelle in un'unica istruzione DIM, separandoli con virgole.

Esempio:

```
10 DIM V(15)
20 DIM M(20,20)
```

Le indicazioni espresse dalle due DIM potevano essere fornite usando un'unica istruzione DIM in questa forma:

```
10 DIM V(15),M(20,20)
```

In questo esempio è riservato uno spazio di 16 numeri per il vettore V:

V(0), V(1), V(2),...V(15)

e di 441 numeri per la matrice M:

20 + 1 righe e 20 + 1 colonne

Istruzioni e funzioni avanzate

Questa sezione si propone di trattare brevemente le caratteristiche più avanzate e specializzate del BASIC. Queste caratteristiche, contrariamente alle istruzioni e funzioni elementari di cui si è già parlato, non si trovano necessariamente in ogni versione di BASIC. Per essere certi che le funzioni e/o le istruzioni desiderate siano disponibili, bisogna consultare il manuale di programmazione del particolare interprete del BASIC con cui si opera.

Istruzione INPUT

L'istruzione INPUT agisce come l'istruzione READ, tranne che l'input dei dati avviene dalla tastiera di console, invece che tramite l'istruzione DATA. Ogni volta che il BASIC incontra l'istruzione INPUT, stampa un punto interrogativo (?), per indicare che è in attesa di input da operatore . Allora l'utente batte l'input desiderato (così come si presenta normalmente in una istruzione DATA) terminando col ritorno carrello. Generalmente le istruzioni INPUT sono combinate con le istruzioni PRINT per indicare quale valore è desiderato:

```
10 PRINT "NUMERO DI GIORNI"".......";
20 INPUT G
```

Il punto e virgola alla fine della istruzione PRINT fa sì che il punto interrogativo sia stampato sulla stessa linea del messaggio. Normalmente, il punto interrogativo è stampato su una linea a parte.

In questo esempio, le parole NUMERO DI GIORNI, senza virgolette, sono stampate seguite da un punto interrogativo. Il numero battuto, seguito dal ritorno del carrello, è assegnato alla variabile G.

Il dato che è stato fornito tramite una istruzione INPUT non è salvato con il programma, perciò questa istruzione andrebbe usata soltanto quando sono fornite piccole quantità di dati, o un dato che non è conosciuto a priori.

Istruzione RESTORE

L'istruzione RESTORE tipicamente permette che i dati forniti tramite le istruzioni DATA vengano letti più di una volta.

Tutte le volte che si incontra una istruzione RESTORE, il BASIC riporta il puntatore della zona dati alla prima posizione del blocco dati. Allora la prima istruzione READ che si incontra riprende a leggere dall'inizio il blocco dati.

Funzione SIGN

La funzione

SGN(X)

fornisce un valore indicante il segno dell'argomento specificato. Ai numeri positivi è assegnato il valore 1, 0 allo 0, -1 ai numeri negativi.

Esempio:

SGN(2,75) = 1
SGN(0) = 0
SGN(-0,25) = −1

Questa funzione può essere combinata con l'istruzione ON, fornendo così tre possibili ramificazioni, dipendenti dal segno del numero:

```
60 ON SGN(X) +2, GO TO 100, 200, 300
```

Questa istruzione farà saltare il programma alla linea 100 se X è negativo, a 200 se X è uguale a 0, a 300 se X è positivo.

Funzione INTEGER

La funzione

INT(X)

esprime il massimo numero intero di X, che è minore o uguale a X.

Esempio:

INT(2,98) = 2
INT(-2,05) = −3
INT(4) = 4

Questa funzione può essere usata per arrotondare un numero all'intero più prossimo, sommando al numero il valore 0,5:

INT(X+0,5)

Può essere usata anche per troncare un numero nella cifra decimale ennesima:

INT(X*10**n +0,5) /(10**n)

Numeri casuali

Molte forme di BASIC hanno la possibilità di generare un numero casuale compreso fra 0 e 1. Questa funzione viene generalmente usata per simulare eventi che si verificano in modo alquanto casuale, come i punti in una partita a dadi.

La forma generale di questa funzione è:

RND

senza argomento.

Se l'argomento è richiesto da una particolare versione di BASIC, si usa generalmente il numero 1:

RND(1)

Per generare un numero intero casuale di una sola cifra, si può usare l'istruzione seguente:

```
40 LET X=INT(10*RND)
```

Se si vuol generare un numero intero casuale com-

preso fra due limiti, si può usare questa istruzione:

60 LET X=INT(A*RND+B)

Questa istruzione genererà un numero intero casuale compreso fra A e B, dove A è maggiore di B.

Così, volendo simulare un lancio di dadi (caso di numeri interi casuali compresi fra 1 e 6) verrà usata l'istruzione:

INT(6*RND+1)

In fase di «debug» di un programma, la tipica funzione di BASIC RND genera la stessa serie di numeri casuali nello stesso ordine, ogni volta che il programma è eseguito.

Al contrario in alcune versioni, inserendo una istruzione RANDOMIZE come prima istruzione in un programma che usa numeri casuali, si otterranno risultati diversi per ogni esecuzione del programma stesso.

Funzioni definite da utente

Oltre alle funzioni standard del BASIC, alcuni interpreti BASIC permettono fino a 26 funzioni supplementari da definirsi con l'istruzione DEF.

Il nome della funzione definita è sempre di 3 lettere, di cui le prime 2 sono sempre «FN» (Es.: FNA, FNB,... FNZ); l'istruzione DEF può comparire in qualunque punto del programma.

Ogni istruzione DEF definisce una singola funzione, e può contenere qualsiasi combinazione di altre funzioni e/o variabili, oltre a quelle che denotano gli argomenti della funzione. Ogni variabile che non sia un argomento di funzione userà il suo valore corrente nel programma.

Ad esempio, volendo usare ripetutamente la funzione

$$e^{X^2} + 3X + Z$$

si definisce la funzione

DEF FNE(X) = EXP(X**2) +3X+Z

e poi si richiedono i differenti valori della funzione con:

FNE(2), FNE(A+B), ecc.

Il valore corrente di Z è usato ogni volta che si chiama la funzione.

Inoltre, ogni funzione definita può avere, come argomento della funzione, zero, una, due, o più variabili numeriche.

Esempi:

```
10 DEF FNA = 31,16*(R**2)
20 DEF FNB(X,Y) = (X+Y)/(X Y)
30 DEF FNR (A,B,C,D) = FNB (A,B) + FNB(C,D)
```

Le funzioni definite su più linee si costruiscono omettendo il segno «=» e terminando la definizione della funzione con una istruzione FNEND.

Nell'ambito di questo tipo di funzioni può essere usato il nome della funzione senza argomento, come variabile provvisoria per calcolare il valore della funzione.

Inoltre, le istruzioni GOSUB e RETURN non sono normalmente permesse in una funzione definita su più linee; in maniera analoga non sono permesse funzioni inserite l'una nell'altra o ancora l'uscire dal gruppo di linee che definiscono la funzione in un punto diverso da FNEND.

Esempio:

```
10 DEF FNM(X,Y)
20 LET FNM = X
30 IF Y = X, THEN 50
40 LET FNM = Y
50 FNEND
```

L'esempio calcola il maggiore dei due valori dell'argomento ogni volta che la funzione è chiamata.

Vettori e Matrici

Poichè ricorrono di frequente operazioni su elenchi e tabelle, nel BASIC è di solito disponibile una speciale serie di istruzioni e di funzioni che operano sulle matrici.

In Tabella 4 è dato l'elenco delle tipiche istruzioni e funzioni MAT, più una breve descrizione delle loro operazioni.

Molte istruzioni di matrice richiedono speciali condizioni per essere legali, o per poter permettere operazioni su vettori o su matrici.

Quando un vettore ha un componente uguale a 0, e una matrice una colonna e una riga uguali a 0, le istruzioni MAT non possono essere utilizzate.

Una serie numerica trattata da una istruzione MAT, che non sia MAT INPUT, sarà elaborata come se fosse una matrice, a meno che non sia dichiarata specificamente come vettore in una istruzione DIM.

Istruzione	Descrizione
MAT READ a,b,c...	Legge le matrici o i vettori specificati, e di dimensioni precedentemente specificate.
MAT M = ZER	Pone tutti i componenti della matrice M uguali a 0.
MAT M = CON	Pone tutti i componenti della matrice M uguali a 1.
MAT PRINT a,b,c...	Stampa le matrici o i vettori specificati.
MAT INPUT V	Richiede l'input di un vettore.
MAT B = A	Pone la matrice B uguale alla matrice A.
MAT C = A+B	Pone la matrice C uguale alla somma delle matrici A e B.
MAT C = A-B	Pone la matrice C uguale alla differenza della matrice A meno la matrice B.
MAT C= A*B	Pone la matrice C uguale al prodotto della matrice A per la matrice B.
MAT C = TRN(A)	Pone la matrice C uguale alla trasposizione della matrice A.
MAT C = (k) *A	Pone la matrice C uguale alla matrice A moltiplicata per un numero k che deve essere in parentesi e può essere dato da una espressione.
MAT C = INV(A)	Pone la matrice C uguale all'inverso della matrice A.

Tabella 4 — Funzioni speciali di matrice in BASIC.
L'aggiunta di queste funzioni di matrice al BASIC è uno dei più comuni ampliamenti delle sue prestazioni. Per le applicazioni scientifiche queste funzioni sono una necessità reale.

Insiemi alfanumerici

Fin qui, sono state discusse soltanto le operazioni che riguardano l'insieme numerico; ma molte forme di BASIC elaborano informazioni alfanumeriche nella forma di insiemi alfanumerici.

State pensando ad un microcomputer?

Assicuratevi di esaminare le offerte di prodotti della più grande Ditta del mondo, produttrice di una linea completa di Microelaboratori.
Tutte le macchine della Ohio Scientific sono dotate del più veloce e completo Basic-In-Rom o Su- Disco, ad impiego istantaneo, esistente nel campo della microelaborazione.

Serie Challenger I

Sistemi di elaborazione economici che parlano in Basic. Ideale per studenti, computer amatori, la scuola, la casa.
SuperBoard II - Il primo sistema al mondo completo, su una scheda comprendente tastiera, interfaccia per video display e audio-cassetta, BASIC-IN-ROM e fino a 8 K di memoria RAM.
Challenger IP - Superboard II completo di contenitore e con alimentatore incorporato.
Challenger IP - Disk - Sistema completo di mini-dischetti espandibile fino a 32 K di memoria RAM.

Serie Challenger II P

Microelaboratori, BUS orientati, ad altissime prestazioni, per impieghi personali, scolastici, di ricerca e nelle piccole aziende.
C2 - 4 P - Il portatile professionale
C2 - 8 P - La macchina personale più espandibile del mondo per impieghi commerciali e di ricerca.
C2 - 4 P - A mini - dischetti il meglio fra i portatili.
C2 - 8 P - A due dischetti per piccola azienda a un prezzo imbattibile.

Serie Challenger II con interfaccia seriale

Ha le stesse grandi caratteristiche della serie Challenger II P per coloro che hanno terminali seriali, utilizzata da piccole aziende, scuole, industrie.

Challenger III il meglio e il più potente nel campo dei microelaboratori

L'unico sistema a 3 processori (6502, 6800, Z80) per soddisfare le più sofisticate esigenze delle imprese
C3 - OEM - Il sistema a due dischetti compatto per grandi volumi di lavoro con 32 K di memoria RAM.
C3 - B - Il sistema basato sul disco Winchester da 74 M bytes. Il microelaboratore più potente del mondo.
La Ohio Scientific vi offre anche la più ampia gamma di accessori per l'espansione e la più grande scelta di software economico.
Confrontate un modello della Ohio Scientific con un modello comparabile di un'altra Casa che avete preso in considerazione. Confrontate le prestazioni, la reale possibilità di espansione, il software e i prezzi, e capirete perchè siamo diventati la più grande società di microelaboratori più estesa del mondo che produce la linea di sistemi.

PRODUTTORE

OHIO SCIENTIFIC

DISTRIBUTORE INTERNAZIONALE

AMERICAN DATA-HOME AND OFFICE COMPUTER

DISTRIBUTORE ESCLUSIVO PER L'ITALIA

EDICONSULT s.r.l.

OHIO SCIENTIFIC

DISTRIBUTORE

SKILAB

Sono interessato ai Microelaboratori OSI.
Inviatemi informazioni su:

- **Elaboratori personali**
- **Sistemi scolastici**
- **Elaboratori per piccole aziende**
- **Sistemi di sviluppo industriale**

Distributore esclusivo per l'Italia:

EDICONSULT s.r.l.

Via Caccini, 12 - 20052 MONZA - Tel. 039/389850

Nome ....

Indirizzo ....

Tel. ....

Città .... **CAP** ....

Un insieme alfanumerico è una sequenza di caratteri che comprende lettere, cifre, spazi, o altri caratteri di stampa. Il BASIC, tuttavia, non può trattare i caratteri di avanzamento linea, ritorno carrello, avanzamento di forma, o tabulazione verticale.

I caratteri di un insieme alfanumerico sono di norma chiusi fra virgolette (Es.: «RISPOSTA»), quando sono costanti (ma in alcuni casi le virgolette si possono omettere).

Un insieme alfanumerico può essere formato da variabili semplici o da vettori, ma non da matrici. Ogni variabile o nome di vettore seguito dal segno di dollaro ($) sta a significare un insieme alfanumerico.
Es.: A$, B2$, o C$(1)

Ne sono un esempio le prestazioni di PEEK e POKE, che sono usate dal BASIC dell'Altair per manipolazioni a livello di bit.

Come ho già detto più volte, la cosa migliore è sempre consultare il manuale di programmazione relativo alla particolare versione di BASIC usata, controllando sempre quali caratteristiche sono offerte, e come vanno usate, dal momento che possono cambiare radicalmente da sistema a sistema. Inoltre il sistema usato può avere anche speciali comandi operativi e di «editing».

Concludendo...

Ora che sono state trattate le istruzioni e le funzioni

ASC (singolo carattere o mnemonico di carattere non stampabile)
Fornisce il valore decimale corrispondente al codice ASCII dell'argomento.

CHR$ (espressione numerica)
Funzione opposta ad ASC. L'argomento è troncato a un numero intero che è interpretato come carattere alfanumerico.

INSTR (espressione numerica, espressione alfanumerica, espressione alfanumerica)
Ricerca la sequenza alfanumerica data dalla seconda espressione entro la prima. La ricerca inizia dalla posizione di carattere specificata dalla espressione numerica troncata ad un numero intero o dal primo carattere, se questa è omessa. Fornisce la posizione del primo carattere nella prima espressione alfanumerica se lo trova, altrimenti O.

LEFT$ (espressione alfanumerica, espressione numerica)
Fornisce un sottoinsieme alfanumerico della prima espressione, partendo da sinistra. Il sottoinsieme contiene il numero di caratteri specificato dalla espressione numerica troncata ad un numero intero.

LEN (espressione alfanumerica)
Fornisce il numero di caratteri del suo argomento.

MID$ (espressione alfanumerica, espressione numerica, espressione numerica)
Fornisce un sottoinsieme della espressione alfanumerica partendo dalla posizione di carattere specificata dalla prima espressione numerica troncata ad un numero intero. Il sottoinsieme contiene il numero di caratteri specificato dalla seconda espressione numerica troncata ad un numero intero, oppure tutta la parte restante se la seconda espressione numerica è omessa.

RIGT$ (espressione alfanumerica, espressione numerica)
Fornisce un sottoinsieme della espressione alfanumerica (partendo da destra) formato dal numero di caratteri specificato dalla espressione numerica troncata ad un numero intero.

SPACE$ (espressione numerica)
Fornisce una sequenza di spazi di lunghezza specificata dalla espressione numerica troncata ad un numero intero

STR$ (espressione numerica)
Fornisce una rappresentazione alfanumerica del suo argomento (il numero è convertito in una sequenza alfanumerica).

VAL (espressione alfanumerica)
Ha funzione opposta a STR$. Fornisce il numero rappresentato dalla espressione alfanumerica (cioè questa è convertita in un numero).

Oltre a queste funzioni, si usa l'istruzione CHANGE per trasformazioni da dati alfanumerici a dati numerici e viceversa:

CHANGE |espressione alfanumerica| TO |espressione numerica|

cambia l'espressione alfanumerica specificata in un vettore numerico. L'elemento 0 del vettore conterrà il numero di caratteri della espressione alfanumerica; il codice decimale corrispondente all'ASCII delle lettere sarà memorizzato nel vettore numerico.

L'istruzione:

CHANGE |vettore numerico| TO |variabile alfanumerica|

cambia il vettore numerico in una sequenza alfanumerica.

L'elemento 0 del vettore deve contenere il numero di caratteri che si desidera nella sequenza.

Tabella 5 — «Tipica» serie di funzioni per la manipolazione di insiemi alfanumerici implementati in un interprete BASIC.

La tabella elenca funzioni che sono state trovate nel corso delle esperienze compiute dall'autore con il BASIC su computers della Digital Equipment Corporation; una serie analoga di funzioni è spesso implementata da altri interpreti «extended BASIC».

In molte istruzioni del BASIC le variabili o le costanti alfanumeriche sono usate come se fossero quantità numeriche. Tuttavia, i dati numerici e i dati alfanumerici sono memorizzati in due distinte zone, e sono utilizzati indipendentemente gli uni dagli altri.

L'istruzione RESTORE riporta gli indici delle due serie di dati all'inizio delle rispettive zone dati. Un tipico accorgimento di implementazione per azzerare solo l'indice dei dati numerici consiste nell'usare RESTORE*, e RESTORE$ per azzerare solo l'indice dei dati alfanumerici.

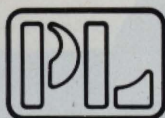
La Tabella 5 mostra alcune delle funzioni per manipolare gli insiemi alfanumerici in BASIC.

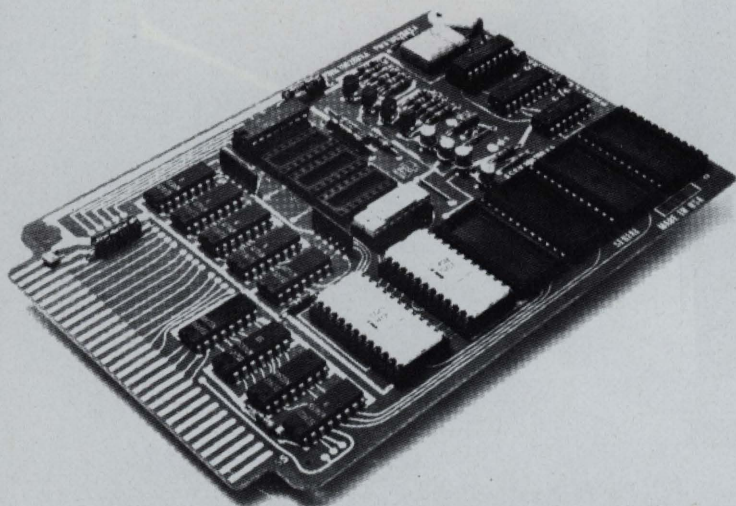
Altre caratteristiche

Molti sistemi che utilizzano il BASIC, specialmente quelli che lavorano in «time-sharing», a volte permettono speciali operazioni sui flussi dati.

generali e alcune delle caratteristiche più avanzate, il sistema migliore per imparare veramente il linguaggio BASIC è l'esperienza pratica. Una volta trovato un sistema con BASIC, occorre controllarne le caratteristiche, e a questo punto si può partire con programmi piccoli e semplici, senza pensare di essere subito capaci di programmare tutto quello che si vuole, o quasi.

Il BASIC è un linguaggio estremamente facile sia da imparare, sia da usare, quindi non bisogna lasciarsi spaventare dall'espressione «LINGUAGGIO AD ALTO LIVELLO».

Nei prossimi articoli sulle tecniche di programmazione e le applicazioni, si vedrà come il BASIC diventa disponibile su un numero sempre maggiore di sistemi a microcomputer. I programmi in BASIC sono molto facili da riprodurre e da convertire sui vari sistemi specifici, compresi i microcomputers.

Pro-log

AVETE UN MICROPROCESSOR? PRO-LOG HA TUTTO IL RESTO: SUBSISTEMI - SCHEDE PRECABLATE PROGRAMMATORI - DOCUMENTAZIONE

Quanto vi costerebbe progettare e costruire un sottosistema come il ns. PLS-881? Esso comprende:

- MICROPROCESSOR INTEL 8080A
- CLOCK
- RESET ESTERNO E INTERNO
- SOCKETS PER 4096 BYTES PROM
- RAM DA 1024 CARATTERI
- LOGICA DI I/O (16 INGRESSI TTL; 24 USCITE TTL; 4 USCITE MOS)

TUTTO SU UNA SOLA SCHEDA AL COSTO SEGUENTE:

QUANTITÀ	PREZZO
1- 9	$ 260
10- 24	$ 210
25- 99	$ 185
100-249	$ 165

MA FORSE AVETE BISOGNO DI UN SISTEMA PIÙ POTENTE

In tal caso abbiamo sistemi su 2,3 schede basati su chip e 4040 o sistemi su 1,3 o 5 schede basati su chip 8080A, 8085 6800 e Z 80. E inoltre:

PROGRAMMATORE M 900 - DUPLICATORE M 920

Tutti possono programmare e duplicare **PROM** bipolari o **MOS** direttamente tramite tastiera, lettore di nastro o calcolatore remoto. Il programmatore M 900 è dotato di tutte le interfacce necessarie. Inoltre l'impiego di differenti **MODULI PLUG-IN** consente l'adattamento dell'unità virtualmente ad ogni PROM in commercio.

BUFFER RAM

Il programmatore **M 900** può essere dotato di un buffer da **1 - 2 - 4 K.**

MODULI GENERICI

A basso costo potrete programmare con un solo modulo plug-in **un'intera famiglia di componenti.**

VISITATECI AL BIAS
21-25 NOVEMBRE
PAD. 13

Rappresentante esclusivo per l'Italia:

technitron

00197 ROMA - Via Mangili, 20 (Sede)
Tel. 80.56.47-87.24.57 - Telex 68171 - TECRO
20144 MILANO — Via California, 12
Tel. 469 03.12-498.92.79Telex 39252 - TECMI

L'importanza del microprocessore nella scelta del personal computer

di Lou Frenzel - *Heath Company*

I criteri da seguire quando si sceglie un personal computer sono in parte simili a quelli seguiti per la scelta del microprocessore. Il discorso di fondo è, però, abbastanza diverso. L'importanza del fattore software.

La scelta di un personal o di un hobby computer è anche scelta del tipo di microprocessore, dato che ogni personal computer è costruito attorno ad un microprocessore.

I costruttori di semiconduttori hanno messo sul mercato un numero notevole di unità a microprocessore (vedi figura 1) con diversificazioni notevoli nelle prestazioni offerte da ciascuno, per cui è naturale che esista almeno una mezza dozzina di differenti processori nel campo del personal computer. Questa varietà di prodotti sembrerebbe rendere la scelta più elastica, ma in realtà, come ogni scelta, la decisione è più problematica, soprattutto se si è degli esordienti nel campo dei sistemi.

Intel 8080, 8085, 8048, 8086
Motorola 6800,6801
MOS Technology 6502
Zilog Z-80
Signetics 2650
RCA 1802
Fairchild F8, 9440
Mostek 3870
Intersil 6100
Texas Instruments 9900
National Semiconductor SC/MP, PACE, 8900
DEC LSI-11
Data General microNOVA General Instrument 1600

Figura 1 - Elenco dei principali microprocessori e dei loro costruttori.

Questo articolo si propone di fornire indicazioni per una scelta intelligente nel campo del personal computer, mettendo a fuoco i criteri di scelta del miglior microprocessore per il proprio personal computer.

La difficoltà della scelta è evidente se si considerano i numerosi dispositivi a microprocessore attualmente disponibili, ma questo è relativamente poco se poi si tiene conto del fatto che molti di questi dispositivi saranno soggetti a perfezionamenti successivi da parte dei rispettivi costruttori e che i miglioramenti avvengono ad una rapidità incredibile.
È evidente che questo genere di problema non può avere una soluzione completa, cosi come è evidente che in questo campo il perfezionamento dei dispositivi sarà continuo. Per questa ragione la scelta dev'essere fatta nei confronti di quei dispositivi che non solo presentano potenzialmente una «longevità» più grande, ma soprattutto vengono meglio incontro alle proprie esigenze applicative. A tale proposito è opportuno tenere in considerazione anche alcuni dei criteri a cui fa riferimento il progettista, anche se alcuni di essi incidono meno nella scelta del personal computer (ma quasi in pari misura se la scelta è orientata verso un hobby computer).

Costo

Il costo è un elemento che ha un notevole peso nella scelta di un microprocessore. Comunque questo fattore interessa maggiormente il costruttore di computers che l'utente del sistema. Ad ogni modo è opportuno sapere che la maggior parte dei circuiti integrati a microprocessore hanno approssimativamente lo stesso prezzo e che il costo del solo micro-

processore è soltanto una frazione del costo globale dell'intero sistema: il costo della memoria e delle periferiche incide molto più pesantemente.

Velocità

Uno dei fattori presi in considerazione per confrontare le prestazioni dei computers è la velocità di esecuzione, cioè la velocità a cui le singole istruzioni sono eseguite. La velocità è determinata fondamentalmente dalla frequenza di clock; comunque è anche condizionata dal tempo di accesso della memoria e dalla architettura del processore. La maggior parte degli attuali microprocessori non hanno elevate velocità di elaborazione in quanto spesso sono circuiti a semiconduttori a tecnologia MOS e quindi intrinsecamente più lenti dei circuiti bipolari (TTL).

In questi anni sono stati apportati numerosi miglioramenti per incrementare la velocità dei circuiti MOS; i circuiti MOS a «canale P» sono stati gradualmente sostituiti dai più piccoli e più veloci circuiti MOS a «canale N». Il continuo sviluppo nel processo a «canale N» promette ulteriori miglioramenti di velocità.

La velocità è comunque un fattore che ha una influenza minima nella scelta di un personal computer, dato che la maggior parte dei microprocessori a tecnologia MOS esegue una istruzione in parecchi microsecondi, che risulta essere un tempo abbastanza veloce per la maggior parte delle applicazioni. È vero che le velocità di elaborazione variano da uno a quattro tra i vari microprocessori MOS, ma è anche vero che la differenza, in ambito personal computer, non si presta ad essere notata. Considerazioni di velocità devono essere fatte se l'applicazione comporta operazioni matematiche complesse o funzioni da eseguire in tempo reale.

Potenza di calcolo

La potenza di calcolo è un termine piuttosto vago per indicare la potenza del «set» di istruzioni e l'architettura del computer. Questa caratteristica, che dipende anche dalla velocità, è più comprensiva delle capacità di un microprocessore.

Risulta difficile dare delle indicazioni specifiche per determinare se un microprocessore è più potente di un altro, per cui ci si deve limitare a prendere in considerazione alcuni fattori che sono senz'altro significativi.

Questi fattori sono: il numero di istruzioni del «set» di istruzioni, il numero dei registri di lavoro, il numero ed il tipo dei modi di indirizzamento.

I microprocessori che hanno il maggior numero di istruzioni, di registri di lavoro e di modi di indirizzamento sono notevolmente potenti in quanto sono in grado di eseguire operazioni complesse in minor tempo.

Questi fattori sono strettamente legati alla architettura del microprocessore; un microprocessore che ha un'organizzazione dei registri flessibile, vari modi di indirizzamento, può avere prestazioni simili a quelle di un altro microprocessore con velocità superiore ma con una architettura più semplice.

Questa circostanza dimostra più chiaramente come non sia semplice rispondere alla domanda sul tipo di microprocessore più potente, ed evidenzia maggiormente il concetto che molto dipende dal tipo di applicazione: quando una particolare applicazione può essere definita, allora la scelta può essere ottimizzata.

Seconda sorgente

Per seconda sorgente si intende un costruttore diverso da quello originale, che produce lo stesso dispositivo.

Quanto un costruttore di semiconduttori introduce un nuovo componente, questi cerca di affermarsi sulla maggior parte di mercato possibile con varie strategie.

Una delle migliori strategie è appunto quella che parecchi costruttori competano per fare lo stesso dispositivo, e di questa circostanza beneficiano non solo il costruttore originale, ma anche i vari utilizzatori, per i seguenti motivi: sorgenti di prodotto alternative, riduzione di prezzo, approvvigionamento più regolare.

Un modo per determinare la popolarità di un microprocessore è di verificare l'esistenza di una seconda sorgente.

Più seconde sorgenti vi sono, più quel dispositivo è utilizzato e più competitivo è il prezzo.

Pertanto anche questo fattore, per quanto poco tecnico, deve essere preso in considerazione quando si sceglie un microprocessore.

Popolarità

Può sembrare poco serio includere un tale criterio apparentemente generico e senza significato fra le indicazioni da seguire quando si sceglie un microprocessore, eppure anche questa circostanza è importante.

Quanto meno questo fattore indica che quel particolare componente è ben conosciuto e largamente impiegato da altri, e questo fatto può far pensare che ci deve essere più di una ragione perchè sia così. Non bisogna poi trascurare i conseguenti vantaggi di maggior software disponibile.

Comunque anche se un dispositivo non presenta i requisiti di disponibilità di software, di velocità e di potenza di calcolo, ciò non toglie che possa essere preso in considerazione o per particolari valutazioni sul produttore o per sue specifiche caratteristiche. Spesso scegliendo un microprocessore vengono acquistate capacità potenziali piuttosto che pratiche capacità di elaborazione per cui anche se il pensiero di avere il microprocessore più potente, più veloce, più nuovo è una grande tentazione, non è male essere più pratici, più realistici e più analitici nella scelta del microprocessore per il proprio personal computer.

Documentazione

Documentazione è tutto ciò che di scritto è disponibile per un particolare microprocessore, intendendo non solo il materiale fornito dal costruttore, ma anche articoli di riviste, libri, ecc.

La buona documentazione determina il successo del proprio lavoro, fornisce un valido aiuto nelle applicazioni, facilita l'apprendimento e l'utilizzo dei vari dispositivi e non solo per i principianti.

In conclusione, bisogna considerare sempre questo fattore prima di fare la scelta finale.

Compatibilità

La compatibilità concerne il futuro di un dato microprocessore e indica che quando un microprocessore è migliorato o sostituito da un altro dispositivo, questo sarà, a livello software, compatibile col precedente.

I costruttori di computers notarono subito che la compatibilità verso i prodotti di una medesima linea era un punto di estrema importanza per il proprio sviluppo e per la propria strategia di marketing.

Infatti chi acquista un computer sviluppa poi da sè il software applicativo. Se in seguito decide di cambiare computer, prende in considerazione solo computers compatibili con quello sostituito in modo che il software sviluppato possa essere eseguito sul nuovo computer.

Diversamente se viene scelto un microprocessore o computer completamente diverso, diventa necessario o abbandonare il software sviluppato, o convertirlo, e l'una e l'altra circostanza non è desiderabile in quanto significa dover affrontare un investimento considerevole.

Con riferimento ai microprocessori la scelta di un dispositivo che presenta la possibilità di essere sostituito o migliorato con uno compatibile è senz'altro una buona scelta. Questo significa utilizzare vantaggiosamente l'attuale sistema ed in seguito acquisire un sistema più potente senza perdere tempo e denaro per lo sviluppo di nuovo software.

Software

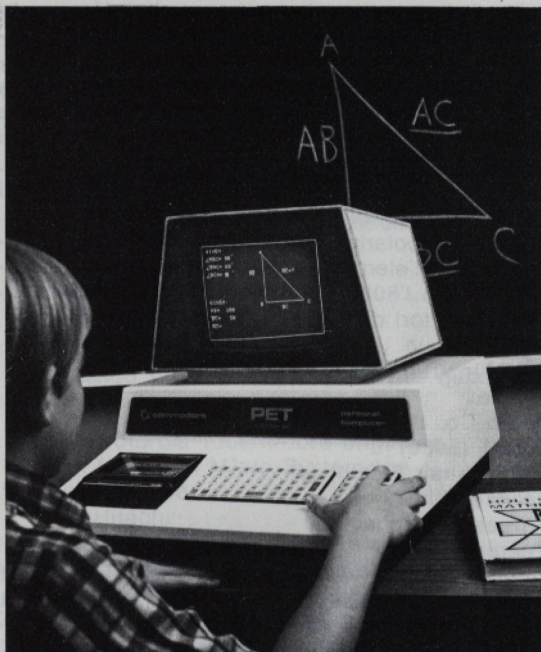
Sarebbe quasi superfluo menzionare l'importanza del fattore software nella scelta di un microprocessore.

Anche chi è agli inizi sa che l'hardware di un microprocessore è inutilizzabile senza un «buon» software.

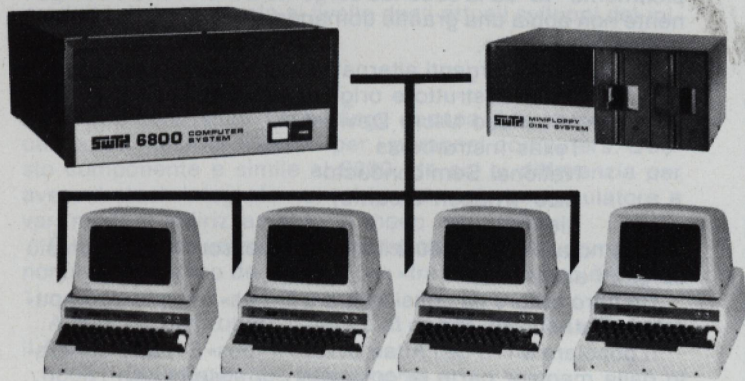
HOMIC

Presenta in Italia i computer personali

COMMODORE PET E RADIO SCHACK TRS-80

I PERSONALI ALL'AVANGUARDIA

- Per la scuola
- Per il laboratorio
- Per il Club

- Per la casa
- Per lo studio professionale
- Per la piccola impresa

SWTPC 6800 il potente microsistema operante in time-sharing

- Per la gestione di piccole-medie aziende
- Per la istruzione programmata nella scuola e nei laboratori linguistici
- Per lavori scientifici

NASCOM Z80 l'economico sistema in KIT operante in assembler e basic

per
- l'industria
- la scuola
- l'hobbystica

HOMIC

Bottega di dimostrazione: P.za de Angeli 3
Ufficio: via Dante, 9 Milano tel 809456

Bisogna considerare non solo il software di sistema necessario allo sviluppo dei programmi applicativi, ma anche la disponibilità di programmi sviluppati da altri e che possono essere eseguiti sul proprio computer.

La maggior parte degli utenti di personal computers desidera scrivere e sviluppare da sè i propri programmi, ma bisogna considerare che il valore di un sistema è più alto se è possibile disporre facilmente di software.

Fra tutti i fattori finora considerati, quello relativo al software ed alla sua disponibilità è senz'altro il più determinante nella scelta di un microprocessore.

Sul software devono essere fatte due considerazioni: la prima concerne la facilità di apprendimento del funzionamento del microprocessore e della relativa programmazione; la seconda, già accennata, riguarda la disponibilità di software già sviluppato.

Nel primo caso, la semplicità del «set» di istruzioni e della architettura è determinante per l'utilizzo di un microprocessore, dato che, se il «set» di istruzioni è di facile concezione e l'architettura semplice, altrettanto facile e semplice sarà la programmazione.

Anche chi è agli inizi non solo imparerà ad usare rapidamente un microprocessore, ma soprattutto otterrà presto risultati soddisfacenti.

Nel secondo caso (disponibilità di software), il vantaggio di un microprocessore molto popolare e largamente utilizzato consiste nel poter accedere ad una enorme quantità di programmi, sia perchè è possibile trovare articoli su riviste specializzate con «listing» di programmi relativi a quel microprocessore sia perchè è possibile acquistare direttamente «package» di software.

La scarsa disponibilità di software è una chiara indicazione che quel processore non è largamente utilizzato, per cui l'eventuale utente avrà a suo carico lo sviluppo della maggior parte di software.

Per concludere, è bene ricordare che questo fattore è quello che di gran lunga supera, come importanza e come criterio di scelta, i fattori di velocità, costo e potenza di calcolo.

I quattro «Big»

Fra tutti i microprocessori elencati in precedenza, quattro sono quelli più popolari e più utilizzati; è una stima molto sicura quella che attribuisce a questi componenti il 90% del mercato dei microprocessori nei personal computers.

I microprocessori più largamente utilizzati nei personal ed hobby computers sono nell'ordine:

8080, 6800, 6502 e Z-80.

Per ciascuno di questi è disponibile una notevole quantità di software ed è inoltre garantita la compatibilità.

È opportuna, pertanto, l'analisi di questi componenti per illustrare le loro capacità e le loro future potenzialità.

8080-8085

Il microprocessore 8080 dell'Intel fu il primo della seconda generazione di microprocessori a 8 bit, e appunto per questo si impose facilmente nel mercato dei processori a 8 bit.

I successivi microprocessori della seconda generazione come ad esempio il 6800 hanno avuto bisogno di maggior tempo per imporsi sul mercato proprio a causa dell'enorme successo dei prodotti Intel.

L'8080 fu annunciato nel 1973 ed ancor oggi, nonostante l'invasione da parte di altri componenti a 8 bit, è il più diffuso.

Benchè per architettura, velocità e potenza di calcolo, l'8080 non sia spettacolare soprattutto se paragonato ad altri chips, tuttavia questo componente è stato ed è utilizzato in applicazioni industriali e in sistemi a microcomputers con caratteristiche «general-purpose». È inoltre supportato da una buona documentazione; soprattutto esiste più software per l'8080 che per qualsiasi altro microprocessore a 8 bit.

Come stima, abbastanza esatta, si può dire che il 60% di tutti i microprocessori a 8 bit utilizzati sono 8080.

Un altro fattore di merito dell'8080 è che è garantita la compatibilità verso i componenti della medesima linea.

Il nuovo microprocessore dell'Intel, cioè l'8085, è un 8080 migliorato, per cui utilizzando un 8085 è possibile sviluppare un microcomputer con maggiori prestazioni.

L'8085 ha un numero minore di chips di supporto in quanto le funzioni di clock e di controller (separate nell'8080), sono invece integrate nel chip 8085. In più l'8085 richiede una singola alimentazione invece che le due richieste per l'8080. Infine l'8085 ha una più alta velocità di esecuzione ed un maggior numero di istruzioni.

Un altro dispositivo compatibile con l'8080 è l'altrettanto ben noto Z-80. Questo microprocessore, molto più recente dell'8080, è più potente e con maggiori capacità. Lo Z-80 include le istruzioni dell'8080 di modo che il software sviluppato per l'8080 può essere eseguito anche sullo Z-80; in altre parole, il «set» di istruzioni dell'8080 è un «sub-set» delle istruzioni dello Z-80. Più esattamente, lo Z-80 possiede quasi il doppio di istruzioni dell'8080, il che lo rende molto potente. Inoltre anche lo Z-80, come l'8085, richiede pochi chips di supporto ed una singola alimentazione di 5 V.

L'evidenza della popolarità dell'8080 può essere dimostrata semplicemente dall'elenco di costruttori di personal computers che utilizzano l'8080. Un elenco, probabilmente incompleto, di costruttori di sistemi con 8080 comprende:

Digital Group
E & L Instruments
Equinox
Heath Co. (H8)
IMSAI (8080)
MITS (Altair 8800b)
PolyMorphic
Processor Technology
Vector Graphic

Un'altra considerazione è il numero di seconde sorgenti per questo chip. Come è stato indicato prima, il numero di seconde sorgenti evidenzia la popolarità di un particolare microprocessore in quanto un costruttore di semiconduttori tipicamente fa da la seconda sorgente a meno che il componente non abbia una grande domanda e non vi sia un mercato ben identificato.

Un elenco di sorgenti alternative per l'8080 è il seguente:

Intel (costruttore originario dell'8080)
Advanced Micro Devices
Texas Instruments
National Semiconductor
NEC (Nippon Electric)
Siemens

È da notare che l'8080 è il microprocessore a 8 bit con più sorgenti alternative.

Un altro fattore da considerare è il «bus» dei microcomputers basati sull'8080.

Il popolare MITS dell'Altair ovvero il «bus» S-100 è utilizzato dalla maggior parte di costruttori di sistemi con l'8080.

In effetti il «bus» S-100 è un «bus» 8080 in quanto i segnali definiti su questo «bus» sono peculiari dell'8080.

Il «bus» S-100, benchè non sia uno standard ufficiale, ormai è considerato tale, per cui è dato modo all'utilizzatore di fare un'ampia scelta di opzioni e di accessori per il suo microcomputer basato sull'8080.

Comunque è da precisare che il «bus» proposto dalla Altair è diventato di fatto uno standard a causa della sua ampia diffusione per cui molti costruttori lo hanno adottato per il vantaggio di mercato già consolidato che di fatto si presentava.

Benchè il «bus» della Altair non sia certo una scelta ottima, è indubbio però che la sua diffusione rende disponibile una notevole quantità di prodotti compatibili con esso.

In conclusione, scegliendo un sistema con il microprocessore 8080, spesso si sceglie anche il «bus» S-100. Ad ogni modo si tenga presente che esistono sul mercato sistemi che non usano il «bus» S-100: tali sono ad esempio i sistemi Heathkit e Digital Group.

6800

Il secondo più popolare e più diffuso microprocessore è il 6800 della Motorola. Questo componente fu annunciato un anno dopo l'8080 ed è riuscito ad imporsi benchè non sia diffuso come l'8080.

L'architettura del 6800 è estremamente semplice e di tipo classico; il suo «set» di istruzioni è facile da imparare e da capire e inoltre presenta una notevole varietà di modi di indirizzamento.

Benchè sia leggermente più lento dell'8080 e dello Z-80, il 6800 recupera la differenza di velocità con un superiore «set» di istruzioni, col tipo di architettura e con i modi di indirizzamento.

Il software sviluppato per il 6800 è notevole ed è largamente disponibile.

La popolarità del 6800 può essere illustrata, al solito, dal numero di costruttori di personal ed hobby computers che lo hanno utilizzato.

Un elenco probabilmente incompleto, è il seguente:

- Southwest Technical Products (Sw TPC 6800)
- Wavemate
- Electronic Products Associates
- MITS (Altair 680 b)
- Digital Group
- Motorola
- MSI
- Heath Company

Un elenco di seconde sorgenti per quanto concerne il «chip» 6800 è invece:

- Motorola (costruttore originario)
- American Micro System Inc
- Fairchild
- Hitachi

Anche per il 6800 esiste una versione migliorata. Recente è l'annuncio del nuovo 6801 caratterizzato da un maggiore «set» di istruzioni rispetto al 6800 e compatibile a livello software con questo. Inoltre le caratteristiche di maggior velocità, memorie e clock sullo stesso chip, alimentazione singola a 5V ed altre interessanti funzioni portano questo nuovo prodotto della Motorola al livello degli attuali sviluppi dei microprocessori a 8 bit.

6502

Il 6502 della MOS Technology occupa il terzo posto nel campo dei microprocessori per i personal computers. Questo componente è simile al 6800, da cui si differenzia per avere il clock integrato nel «chip», un solo accumulatore e vari modi di indirizzamento in modo indicizzato.

È largamente utilizzato nei personal ed hobby computers nonchè nel campo dei sistemi per «training» (tipico è il KIM-1).

Alcuni dei costruttori di personal computers che utilizzano il 6502 sono:

- Ohio Scientific Instruments
- Apple Computer
- MOS Technology (KIM-1)
- Commodore PET
- Microcomputer Associates JOLT

Le seconde sorgenti del 6502 sono:

- MOS Technology (costruttore originario)
- Synertek
- Rockwell

Come il 6800, anche questo componente è facile da imparare ed usare.

Z-80

Lo Z-80 è uno dei più popolari e certamente uno dei più «chiacchierati» microprocessori a 8 bit di questi ultimi anni.

Benchè sia stato introdotto circa tre anni dopo l'8080 ed il 6800, è riuscito ad imporsi in un'area significativa del mercato dei microprocessori. In particolare è presente largamente nel campo dei personal ed hobby computers.

Dato che lo Z-80 è essenzialmente un 8080 più potente e più perfezionato, può essere considerato come facente parte del «movimento 8080»; nondimeno merita un'attenzione particolare per le sue specifiche peculiarità.

Alcuni fra i personal computers che utilizzano lo Z-80 sono:

- Technical Design Labs
- Cromemco
- Digital Group
- Radio Shack

Le seconde sorgenti dello Z-80 sono:

- Zilog (costruttore originario)
- Mostek
- SGS-ATES

Dal momento che lo Z-80 è in grado di eseguire il software dell'8080, la scelta di un sistema con questo componente è senz'altro eccellente.

Inoltre lo Z-80 ha caratteristiche proprie molto interessanti quali una elevata velocità di esecuzione, un esteso «set» di istruzioni e sofisticati modi di indirizzamento. Lo Z-80 è probabilmente il miglior microprocessore ad 8 bit disponibile oggi sul mercato.

Nonostante questi vantaggi, parecchie delle schede sviluppate con lo Z-80 per essere sostituite nei microcomputers basati sul «bus» S-100, non sono state molto popolari per la ragione che un utilizzatore di un sistema con «bus» Altair (cioè basato su 8080) difficilmente è disposto a spendere una cifra, anche se minima, unicamente per sostituire un processor 8080 con uno Z-80. È vero che la compatibilità software permette di conservare i programmi sviluppati, che lo Z-80 incrementa la potenza di calcolo e la velocità di esecuzione, ma è pur vero che nel campo hobby computer non sono richieste prestazioni di calcolo e di velocità elevate, per cui le caratteristiche meno spinte dell'8080 sono più che adeguate.

La scelta di un microprocessore è un processo molto «sofferto» anche se, una volta che vengano attentamente valutati tutti i fattori precedentemente espressi, il campo è facilmente ristretto. Per un microcomputer con caratteristiche «general-purpose», la maggior parte dei progettisti negli anni '75-'77 faceva presto a orientarsi verso le quattro possibilità descritte in precedenza, benchè poi la scelta fra queste diventasse alquanto soggettiva.

Io in particolare sono stato personalmente coinvolto quanuo venne deciso il tipo di microprocessore per il computer H8 Heathkit. Quando quella scelta fu fatta, si era nel 1975 e lo Z-80 non era disponibile. Il 6502 era un componente ancora nuovo e non vi era alcuna seconda sorgente. Queste circostanze restrinsero la scelta piuttosto rapidamente verso l'8080 e il 6800, anche se questo non aveva ancora raggiunto lo sviluppo attuale, e di conseguenza erano scarsi il supporto di documentazione e la disponibilità di software. A causa di queste circostanze l'8080 divenne per noi la scelta più ovvia, scelta che si è dimostrata in seguito pienamente valida nonostante l'avvenuta crescita del 6800 e l'annuncio del 6502 e dello Z-80.

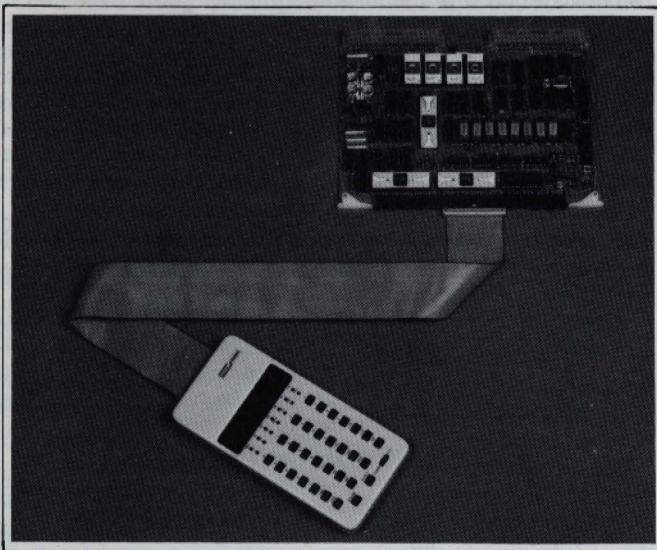
L'8080 ha la potenza di calcolo necessaria alla maggior parte delle applicazioni richieste da un personal o hobby computer.

Oggi comunque una scelta fra Z-80 - 6800 e 6502 è senz'altro valida; la preferenza da accordare ad uno di questi diventa, a questo punto, un fatto abbastanza personale in quanto le considerazioni su capacità tecniche, specifiche e sugli altri fattori precedentemente visti si equivalgono.

Gli altri

È evidente che anche gli altri microprocessori disponibili sul mercato sono capaci di prestazioni uguali se non superiori a quelle dei più diffusi componenti finora considerati, in quanto è stata la scelta fatta dai costruttori di microcomputers a determinare la loro scarsa diffusione nel campo personal computer. Ad ogni modo è opportuno prendere in considerazione anche questi.

HEATHKIT H8 (Business-personal computer).
È l'unità base attorno a cui può essere costruito un sofisticato sistema. Comprende una scheda di CPU, sviluppata attorno all'8080, tastiera a 16 tasti, display a 9 cifre, alimentatore. L'H8 ha un bus esclusivo Heath, completamente bufferizzato, formato da 50 linee.

PET COMMODORE (Business-personal computer)
È un sistema completo. Sviluppato attorno al microprocessore 6502 della MOS Technology, è fornito di interprete BASIC in ROM e di RAM sufficiente per l'esecuzione di programmi complessi. Adotta il bus IEEE della HP.

NANOCOMPUTER SGS-ATES NB Z-80 (Educational-hobby computer). È costituito da una scheda Nanocomputer sviluppata attorno al microprocessore Z-80 e da un terminale, tipo calcolatrice, per l'ingresso e la visualizzazione e la visualizzazione dei dati.

SC/MP

Lo SC/MP della National è un microprocessore a 8 bit molto interessante. È senz'altro uno dei componenti più semplici e più a basso costo attualmente disponibili. La sua semplicità non permette notevole potenza di calcolo, pur tuttavia è estremamente facile da imparare e da usare.

La principale ragione per cui lo SC/MP non è stato ampiamente usato è quella già esposta che nessun costruttore di personal ed hobby computers ha scelto questo componente per i propri sistemi. Comunque la National Semiconductor ha messo a disposizione parecchi «kits» di sviluppo che di fatto hanno destato l'interesse dell'area personal ed hobby computers. In più la National ha sviluppato un linguaggio chiamato NIBL che è un sottolinguaggio BASIC rivolto appunto ad applicazioni con hobby computers.

2650

Il 2650 della Signetics fu annunciato parecchi anni dopo l'8080 e il 6800 e di fatto è ad essi superiore per varie caratteristiche. In effetti è da considerare più un minicomputer che un microprocessore in quanto è notevolmente potente, ha una architettura interessante, un «set» di istruzioni estremamente ricco e infine un'alta velocità di esecuzione.

Eppure questo componente non è riuscito ad essere molto popolare nel campo personal ed hobby computer per la solita ragione di non essere stato scelto dai costruttori di sistemi nonostante le sue evidenti capacità.

F8/3870

Il microprocessor F8/3870 è un componente largamente utilizzato nel campo dei microprocessori a 8 bit. L'F8 è in realtà un microprocessore a due «chips» con la caratteristica di avere una ROM su di un «chip». Il 3870 è la versione Mostek dell'F8 della Fairchild ma realizzata in un unico «chip».

Nessuno di questi componenti è diventato popolare nell'area personal computers, e questo per ragioni intrinseche, dato che questo microprocessore fu progettato non tanto per applicazioni «general-purpose» quanto per utilizzazioni dirette alla sostituzione della logica cablata.

Il discorso sui microprocessori progettati per essere destinati a funzioni di controllore di apparecchiature è qui accennato per segnalare che l'architettura di questi microprocessori è notevolmente diversa dalquella dei microprocessori destinati ad utilizzazioni «general-purpose».

A questo tipo di microprocessori appartengono, oltre all'F8/3870, i nuovi 8048/8748 della Intel. L'8048 come l'F8/3870 incorpora in un unico «chip» molte funzioni; oltre al processore vero e proprio vi sono sul medesimo «chip» clock, ROM, PROM ed i circuiti di interfaccia per operazioni di input/output.

Per concludere, questi microprocessori hanno poche possibilità di essere utilizzati nel campo di computers «general-purpose».

1802

Anche il microprocessore 1802 della RCA è un altro microprocessore a 8 bit interessante. È un componente CMOS e ha quindi un consumo estremamente basso. L'architettura ed il «set» di istruzioni risultano particolari, nondimeno è relativamente facile da utilizzare. Anche per questo componente vale la constatazione che nessun costruttore di personal ed hobby computers lo ha mai utilizzato come processore di un sistema «general-purpose».

Comunque la RCA ha sviluppato parecchi «kits» ideali per iniziare a lavorare sui microprocessori, e uno di questi è finalizzato ad essere un hobby computer a basso costo.

Microcomputers a 12 e 16 BIT

Lo sviluppo dei microprocessori si sta attualmente indirizzando verso componenti più sofisticati e più potenti, per cui se ora il discorso sui microprocessori ruota, per la maggior parte dei casi, attorno a dispositivi a 8 bit, è abbastanza evidente che i microprocessori a 16 bit finiranno col sostituire le unità a 8 bit. Via via che le tecnologie dei semiconduttori miglioreranno, sarà più facile costruire microprocessori a 16

bit così come ora è facile costruire un dispositivo a 8 bit, e quando questo avverrà, la differenza di prezzo sarà minima. Il risultato sarà che i costruttori di nuove apparecchiature si indirizzeranno verso i più potenti componenti a 16 bit anche se la potenza di calcolo di questi è eccessiva per molte applicazioni.

Attualmente vi sono numerosi microcomputers a 16 bit (ed uno solo a 12 bit) che però sono meno diffusi rispetto a quelli a 8 bit, anche se già si nota una crescita del loro impiego via via che nuovi componenti sono sviluppati.

Alcuni dei costruttori di microprocessori a 16 bit sono:

National Semiconductor PACE, 8900
General Instrument 1600
Data General microNOVA
Digital Equipment Corp LSI-11
Texas Instruments 9900
Fairchild 9440

Attualmente i personal ed hobby computers basati su microprocessors a 16 bit sono pochi, e precisamente:

Heathkit H 11 basato sul DEC LSI-11
Technico 9900 basato sul TMS-9900

La ragione per cui i microprocessori a 16 bit non sono molto diffusi nell'area del personal computer è che il loro prezzo è significativamente più alto. Inoltre il software disponibile è ancora scarso.

I microprocessori a 16 bit sono più potenti di quelli a 8 bit, in quanto sono in grado di eseguire elaborazioni di dati in modo più veloce; questa potenza di calcolo è comunque eccessiva per la maggior parte delle applicazioni di personal computer.

Attualmente il più diffuso microcomputer a 16 bit è il DEC LSI11, che deve il suo seguito fra gli hobbysti alla grande quantità di software di base disponibile presso la DEC.

Il primo microprocessore a 16 bit su singolo chip è stato il PACE della National Semiconductor, che però non è stato finora utilizzato dai costruttori di personal computers. Comunque la National Semiconductor ha introdotto per questo componente un sistema di sviluppo a basso costo (PACE LCDS) abbastanza interessante.

Il più nuovo 9900 della Texas Instruments mostra di diventare uno dei più diffusi microprocessori a 16 bit: questo componente può diventare l'8080 dei microprocessori a 16 bit. Il 9900 ha avuto applicazioni in molti campi; è un microprocessore «general-purpose», inoltre molto software della linea minicomputer della Texas Instruments è compatibile con il 9900, e quindi è possibile che in futuro venga convertito completamente per l'utilizzo su questo componente.

Il microNOVA è un altro microprocessore a 16 bit molto potente che è effettivamente supportato dal software disponibile sui minicomputers della linea NOVA della Data General. Questo componente, che è alquanto costoso, ha un'architettura semplice e facile da capire e un supporto di software di base (fornito dalla Data General) enorme. Non è certo però che la Data General voglia promuoverne l'utilizzo nell'area del personal computer o rendere accessibile il proprio software a prezzi competitivi.

Il 9440 MPU della Fairchild è un microprocessore a 16 bit bipolare che usa l'architettura della Data General e può quindi utilizzare il software scritto per queste macchine. Anche questo microprocessore è alquanto costoso.

Il 6100 della Intersil è un microprocessore a 12 bit CMOS e deve la sua fama al fatto che ha l'architettura ed il «set» di istruzioni del famoso minicomputer PDP-8/E della DEC, di cui è in grado di utilizzare il software.

Ma nonostante questo vantaggio il 6100 non ha avuto popolarità nell'area del personal computer soprattutto a causa dell'alto prezzo del sistema basato sul 6100.

MMD-1 (Educational-hobby computer)
Basato sull'8080, è completo di tastiera per la introduzione di dati, e di led per la visualizzazione sia dei dati di ingresso che del contenuto delle celle di memoria. Il sistema è predisposto ad essere espanso.

KIM-1 (Educational-hobby computer).
È sviluppato attorno al microprocessore 6502 della MOS Techology. Ha una tastiera montata sulla scheda, per l'introduzione e l'esecuzione di semplici programmi. Il sistema può essere espanso collegando altri moduli. È possibile, comunque, interfacciare direttamente un registratore di audiocassette.

Conclusioni

Il messaggio di questo articolo è relativamente chiaro. La scelta di un microcomputer per applicazioni nel campo del personal ed hobby computer è ristretta, almeno fino ad ora, ad un limitato campo composto da 8080, Z-80 e 6502, con la probabile «combinazione vincente» di 8080/Z-80.

Ma la più grossa incognita è chi sta facendo «il» microprocessore a 16 bit. Sarà l'8086 della Intel, o lo Z-8000 della Zilog, o il 9900 della Texas Instruments, o qualcun altro?

Infine un altro messaggio è che «il solo processore non fa il computer». Per un utente che programma in BASIC e in PASCAL o in qualche altro linguaggio ad alto livello è quasi irrilevante il tipo di processore. Dopo tutto è il software che produce il «lavoro», per cui la scelta fatta avendo come criteri quelli di possedere un buon sistema e di poter disporre di software è una buona scelta, qualunque sia il livello della architettura della macchina, anche basso.

Lo Z-80 in un sistema multiprocessor

di **Bob Loewer** - *Micro Diversions Inc.*

La struttura multiprocessor permette di risolvere problemi complessi. La soluzione hardware è, però, solo un aspetto del progetto.

Molti progettisti, allo scopo di aumentare la frequenza di trasferimento si sono orientati verso sistemi «multiprocessor».

Tuttavia, con poche eccezioni, queste applicazioni hanno visto il loro campo di impiego limitato al caso di medi e grossi computer causa il loro costo e la loro complessità.

Negli ultimi anni, i microprocessori hanno raggiunto un tale livello di sofisticazione da allargare il loro campo applicativo a sistemi sempre più complessi, offrendo spesso prestazioni paragonabili a quelle di un mini-computer ad un costo estremamente ridotto.

Scopo di questo articolo è la presentazione di un sistema in cui due microprocessori Z-80 operano in una configurazione parallela.

Il microprocessor Z-80

Il microprocessore Z-80, prodotto dalla Zilog, è un dispositivo appartenente alla terza generazione dei componenti LSI, che offre una piena compatibilità software con l'ormai diffuso 8080.

I più interessanti aspetti dello Z-80 riguardano:

un doppio set di registri operativi, un completo insieme di istruzioni (che include la possibilità di manipolazione del singolo bit sia nei registri che in memoria), undici modalità di indirizzamento che includono il tipo indicizzato, generazione interna dell'indirizzo di refresh per l'impiego con memorie dinamiche, un interrupt non mascherabile e una serie di interrupt vettorizzati che consentono un accesso veloce a qualsiasi posizione di memoria.

Dal punto di vista hardware, lo Z-80 presenta un numero ridotto di segnali di controllo che rendono particolarmente semplice il suo inserimento in configurazione «multiprocessor».

Struttura del sistema

L'applicazione presentata consiste in pratica di due microprocessori Z-80 (X e Y) che operano indipendentemente su di una propria memoria di 32 kbytes, e congiuntamente su altri 32K di memoria comune cui entrambi possono accedere, per un totale di 96 kbytes complessivi (vedi fig. 1).

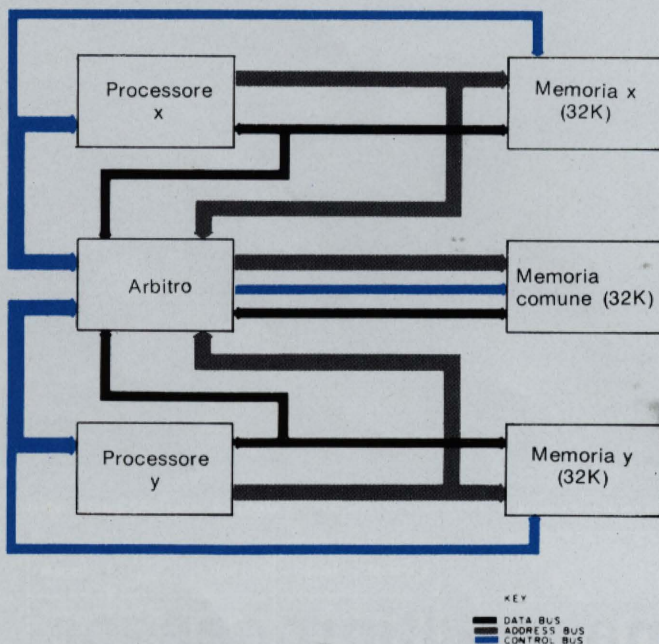



Figura 1 - Il sistema Z-80 in configurazione parallela. Entrambi i processori lavorano indipendentemente ciascuno sulla propria memoria di 32K. La memoria comune, indirizzabile da entrambi i processori come ulteriori 32K, ha propri bus dati ed indirizzi.

La memoria condivisa, dotata di bus dati ed indirizzi distinti dai bus che fanno capo ai due processori, è indirizzabile come area superiore di memoria.

Dati ed indirizzi sono inviati su questi bus quando una delle due CPU si trova in una delle seguenti situazioni:

(1) è in corso una lettura o scrittura di dati nella memoria comune;

(2) si sta eseguendo una fase di fetch nella memoria comune;

(3) si eseguono istruzioni che combinano i casi (1) e (2).

I conflitti di accesso che sorgono quando entrambi i processori indirizzano la memoria comune contemporaneamente, sono risolti da un arbitro, il cui schema è riportato in fig. 2a.

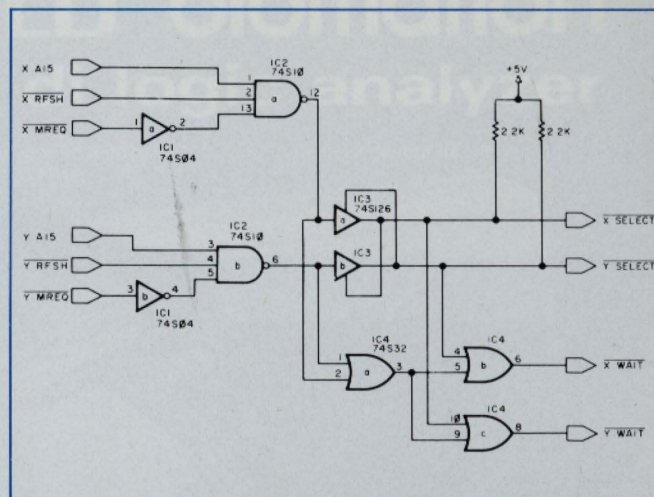



Figura 2a - Logica di arbitraggio della memoria comune. Questo circuito risolve i conflitti che sorgono in caso di accesso contemporaneo al bus della memoria. Una richiesta del processore X ($\overline{XMREQ}$ basso) determina l'attivazione della linea $\overline{XSELECT}$ e disabilita IC3b, bloccando eventuali richieste di Y. Se questa dovesse sopraggiungere mentre X sta utilizzando il bus, IC4 attiverà la linea $\overline{YWAIT}$ forzando Y ad entrare in uno stato di wait.

Poichè i due processori utilizzano opposte fasi del clock, le richieste di accesso al bus non possono mai iniziare al medesimo istante; tuttavia possono ugualmente sorgere dei conflitti in relazione alla sequenza delle istruzioni eseguite. Questo fatto, riassunto in tab. 1, è stato accuratamente studiato per il caso peggiore di conflitto (figura 3): i processori X e Y iniziano simultaneamente una istruzione di lettura o scrittura in cui sia la fase di fetch che quella di esecuzione implicano un accesso alla memoria comune. In tabella 1 sono mostrati tutti i possibili casi di mutuo sfasamento nell'esecuzione dell'istruzione suddetta.

Px/Py Situazione di mutuo sfasamento	Tipo di sincronizzazione M1,T1/M1,T3	M1,T1/M2,T3	Stati di Wait P_X	P_Y	Stati T prima della sincronizzazione
M1,T1/M1,T1		√	0	1	3
M1,T1/M1,T2	√		1	0	3
M1,T1/M1,T3	√		0	0	0
M1,T1/M1,T4	√		0	1	6
M1,T1/M2,T1	√		0	2	7
M1,T1/M2,T2		√	1	0	3
M1,T1/M2,T3		√	0	0	0

Tabella 1 - Analisi della temporizzazione per il sistema a doppia CPU Z-80. Le sette possibili situazioni di mutuo sfasamento tra i due processori sono riportate nella colonna di sinistra. La colonna centrale elenca le due possibili classi di operazioni per CPU in parallelo: la prima (M1,T1/M1,T3) si verifica quando entrambi i processori sono in una fase di fetch di un codice operativo al momento della sincronizzazione; la seconda è il caso di contemporaneità tra il fetch di un codice operativo per il primo processore, ed una operazione di lettura o scrittura in memoria per l'altro. La colonna degli stati di wait indica il numero di stati di attesa di ciascun processore prima della sincronizzazione. Da notare che in due casi l'arbitro non interviene poichè la sincronizzazione è spontanea.

Una istruzione di lettura o scrittura in memoria si completa in sette cicli «T» del clock di sistema; indicando con «M» i cicli macchina, i sette stati attraverso cui evolve l'istruzione sono i seguenti:

Figura 2b - Circuito di controllo della memoria comune. In alto i data bus dei processori X e Y, e della memoria condivisa; a destra il bus indirizzi della memoria; al centro l'arbitro (il cui schema più dettagliato è riportato in figura 2a). Il circuito gestisce le richieste di accesso alla memoria comune secondo il criterio di servirle nell'ordine con cui esse giungono all'arbitro.

M1,T1
M1,T2
M1,T3 } fetch dell'istruzione
M1,T4

M2,T1
M2,T2 } esecuzione della operazione di lettura o scrittura in memoria
M2,T3

Le sette possibili condizioni di sfasamento tra gli stati dei due processori sono date dalla coincidenza dello stato (M1,T1) dell'uno con tutti gli stati da (M1,T1) a (M2,T3) dell'altro.

La figura 3 mostra l'andamento temporale delle richieste di accesso e dei segnali di selezione prodotti dall'arbitro: occorre notare che dopo un periodo di tempo molto breve (sette T al massimo), mediante l'introduzione di uno o due stati di wait, l'arbitro favorisce la sincronizzazione creando completa cooperazione tra i cicli di fetch e di esecuzione dei due processori. Inoltre vi sono due possibili condizioni di sfasamento in cui l'arbitro non interviene poichè i segnali di accesso alla memoria sono già sincronizzati. Prescindendo dalla sequenza delle istruzioni eseguite, questo significa che per l'86% del tempo totale i processori si trovano non sincronizzati di almeno uno stato di wait.

D'altra parte se assumiamo che la sincronizzazione non si perda, entrambi i processori possono eseguire sequenze di istruzioni di lettura o scrittura al 100% di utilizzazione. Certamente questo evento non si verifica per molto a causa delle diverse istruzioni eseguite, tuttavia il criterio di ottenere, nei limiti del possibile, la sincronizzazione è ugualmente il modo più efficiente per risolvere i conflitti di accesso alla memoria.

Logica di arbitraggio

Ogni processore invia all'arbitro tre segnali $\overline{RFSH}$, $\overline{MREQ}$, A15, attraverso i quali formula una richiesta di accesso alla memoria comune (vedi fig. 2b):

$\overline{MREQ}$ indica che è in atto una operazione di lettura o scrittura in memoria;

A15, se posto a livello alto, indica l'area di memoria comune;

$\overline{RFSH}$ è lo strobe di refresh della memoria dinamica (occorre garantire che l'operazione di refresh di una CPU non interferisca con una richiesta di accesso dell'altra).

IC3a e IC3b regolano l'accesso al bus della memorìa in modo mutuamente esclusivo e secondo il criterio di servire le richieste nell'ordine in cui giungono all'arbitro. Una richiesta di X farà sì che IC3a attivi la linea $\overline{XSELECT}$ e contemporaneamente disabiliti IC3b, in modo che il processore Y sia inibito dall'accendere alla memoria per tutta la durata della richiesta di X. Se Y tenta di accedere al bus mentre questo è assegnato a X, IC4 provvederà ad attivare la linea $\overline{YWAIT}$, forzando Y in uno stato di Wait. Quando $\overline{XMREQ}$ va alto indicando che X ha rilasciato il bus, IC3b è abilitato a soddisfare la sopraggiunta richiesta di Y.

La temporizzazione del sistema

La realizzazione di una memoria in grado di servire richieste di accesso sovrapposte da una parte di più processori, implica un accurato esame dei tempi del sistema: parametri critici sono il ritardo introdotto dalla logica di arbitraggio, e l'intervallo di tempo assegnato ad ogni CPU per accedere alla memoria comune.

In un sistema costituito da una sola CPU Z-80, l'intervallo più breve in cui si accede alla memoria si verifica durante il ciclo di fetch di un codice operativo, in cui la durata effettiva della selezione è pari a una volta e mezzo il periodo di clock ($1{,}5\Phi$).

In una configurazione a doppia CPU questo intervallo si restringe per due ragioni:

(1) i ritardi introdotti dalla logica di controllo aggiuntiva;

(2) la sovrapposizione delle richieste di accesso, inevitabile se si vuole sfruttare la piena velocità di elaborazione (fig. 3).

Inoltre il tempo tra due successive selezioni della memoria comune passa da oltre due cicli di clock per il sistema con una sola CPU a meno di un ciclo nella configurazione «dual-processor».

Tutto ciò induce a disporre di una memoria con tempi di accesso e di ciclo più brevi.

La figura 3 mostra come l'intervallo più breve di accesso alla memoria, avente una durata di 1 periodo di clock, si verifichi in caso di conflitto di richieste del bus. In realtà il tempo utile di accesso è ancora più breve, causa i fattori di ritardo introdotti dalla logica di arbitraggio e di decodifica degli indirizzi (vedi Tabella 2).

Quantitativamente, indicando con L_a il tempo di accesso, avremo:

$$L_a = \Phi - d - T_c - T_d$$

dove: Φ è la durata di un periodo di clock, pari a 400 ns;
d è il massimo ritardo del fronte di discesa di $\overline{MREQ}$;
T_c è il ritardo di propagazione attraverso la logica di controllo;
T_d è il ritardo introdotto dalla decodifica degli indirizzi.

Il ritardo T_c tra l'istante in cui $\overline{MREQ}$ diviene attivo e quello in cui si abbassa la linea $\overline{SELECT}$, si può calcolare come segue:

IC attraversati dal segnale	max ritardo (ns)
74S04	5
74S10	5
74S126	18
totale T_c	28 ns

Il ritardo T_d introdotto dalla logica di decodifica, cioè il tempo tra l'attivazione del segnale di $\overline{SELECT}$ e l'effettiva selezione della memoria, si può calcolare facendo riferimento alle figure 2b e 4:

$$T_d = 7408 + \max(74S157_{select};\ 74S157_{enable}) + 74S138$$
$$= 19 + \max(15;\ 11) + 12$$
$$= 46 \text{ ns}$$

Sostituendo nella relazione precedente abbiamo in definitiva:

$$L_a = \Phi - d - T_c - T_d$$
$$= 400 - 20 - 28 - 46$$
$$= 306 \text{ ns}$$

Il risultato ottenuto è largamente sufficiente per una operazione di accesso alla memoria, e non ci obbliga alla scelta di costosi componenti bipolari.

Un altro parametro da considerare è il tempo di ciclo (L_c) della memoria, imposto dalla temporalizzazione del sistema. Questo sarà:

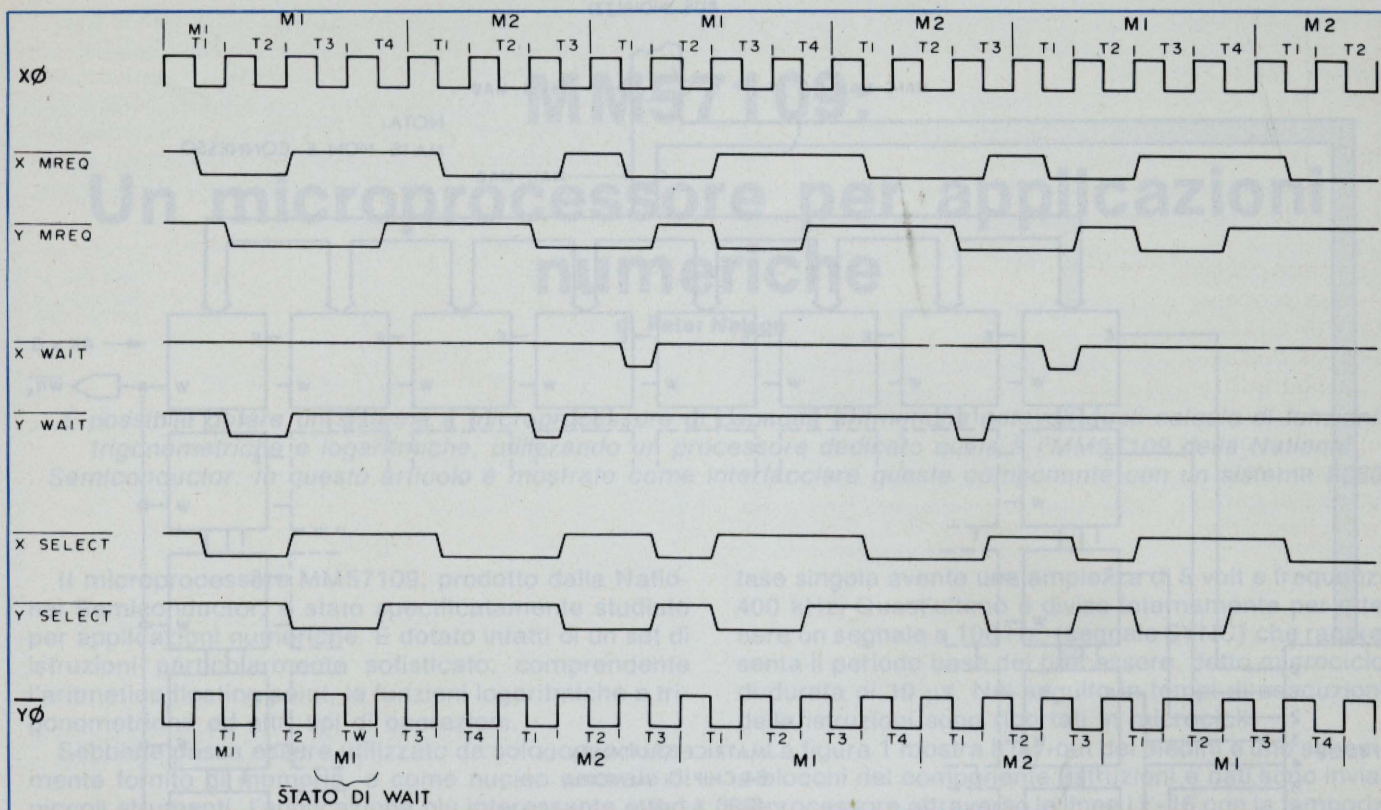



Figura 3 - La condizione M1,T1 /M1,T1 è una delle sette possibili situazioni di mutuo sfasamento mostrate in tabella 1. In questo caso la sincronizzazione è raggiunta dopo tre periodi di clock. (È da notare come l'arbitro attivi periodicamente le linee di WAIT anche durante la sincronizzazione; ciò tuttavia non ha effetto sui processori che campionano il segnale sul fronte di discesa del clock nello stato T2)

Tabella 2 - Timing della logica di arbitraggio. L'arbitro impiega un certo tempo per prendere le decisioni indicate. I ritardi introdotti per le varie transizioni sono mostrati sulla destra.

Condizione iniziale	Condizione finale	Decisioni	Ritardo introdotto (ns)
XA15 alto $\overline{XRFSH}$ alto $\overline{XMREQ}$ basso	a $\overline{XSELECT}$ basso	$\overline{YSELECT}$ alto	28
XA15 alto $\overline{XRFSH}$ alto $\overline{XMREQ}$ basso	a $\overline{XSELECT}$ basso	$\overline{YSELECT}$ basso	25 dopo $\overline{YSELECT}$ va alto
YA15 alto $\overline{YRFSH}$ alto $\overline{YMREQ}$ basso	a $\overline{YSELECT}$ basso	$\overline{XSELECT}$ alto	28
YA15 alto $\overline{YRFSH}$ alto $\overline{YMREQ}$ basso	a $\overline{YSELECT}$ basso	$\overline{XSELECT}$ basso	25 dopo $\overline{XSELECT}$ va alto
XA15 alto $\overline{XRFSH}$ alto $\overline{XMREQ}$ basso	a $\overline{XWAIT}$ basso	$\overline{YSELECT}$ basso	53
YA15 alto $\overline{YRFSH}$ alto $\overline{YMREQ}$ basso	a $\overline{YWAIT}$ basso	$\overline{XSELECT}$ basso	53
$\overline{XSELECT}$ alto	a $\overline{YWAIT}$ alto	$\overline{YSELECT}$ basso	22
$\overline{YSELECT}$ alto	a $\overline{YWAIT}$ alto	$\overline{XSELECT}$ basso	22

$$L_c = \Phi - d$$
$$= 400 \quad 20$$
$$= 380 \text{ ns}$$

È ragionevole calcolare questi tempi utilizzando i valori massimi forniti dal costruttore anziché quelli tipici, e quindi introdurre un ulteriore margine di sicurezza.

Le considerazioni fatte inducono alla scelta di una memoria avente le seguenti caratteristiche:

* Tempo di accesso 280 ns
* Tempo di ciclo 350 ns

Conclusioni e possibili applicazioni

Il principale vantaggio derivante dall'impiego di due (o più) processori operanti in parallelo, per realizzare funzioni complementari è la sensibile riduzione dei costi.

Consideriamo ad esempio un sistema di trasmissione dati configurato a rete: gli elementi nodali realizza-

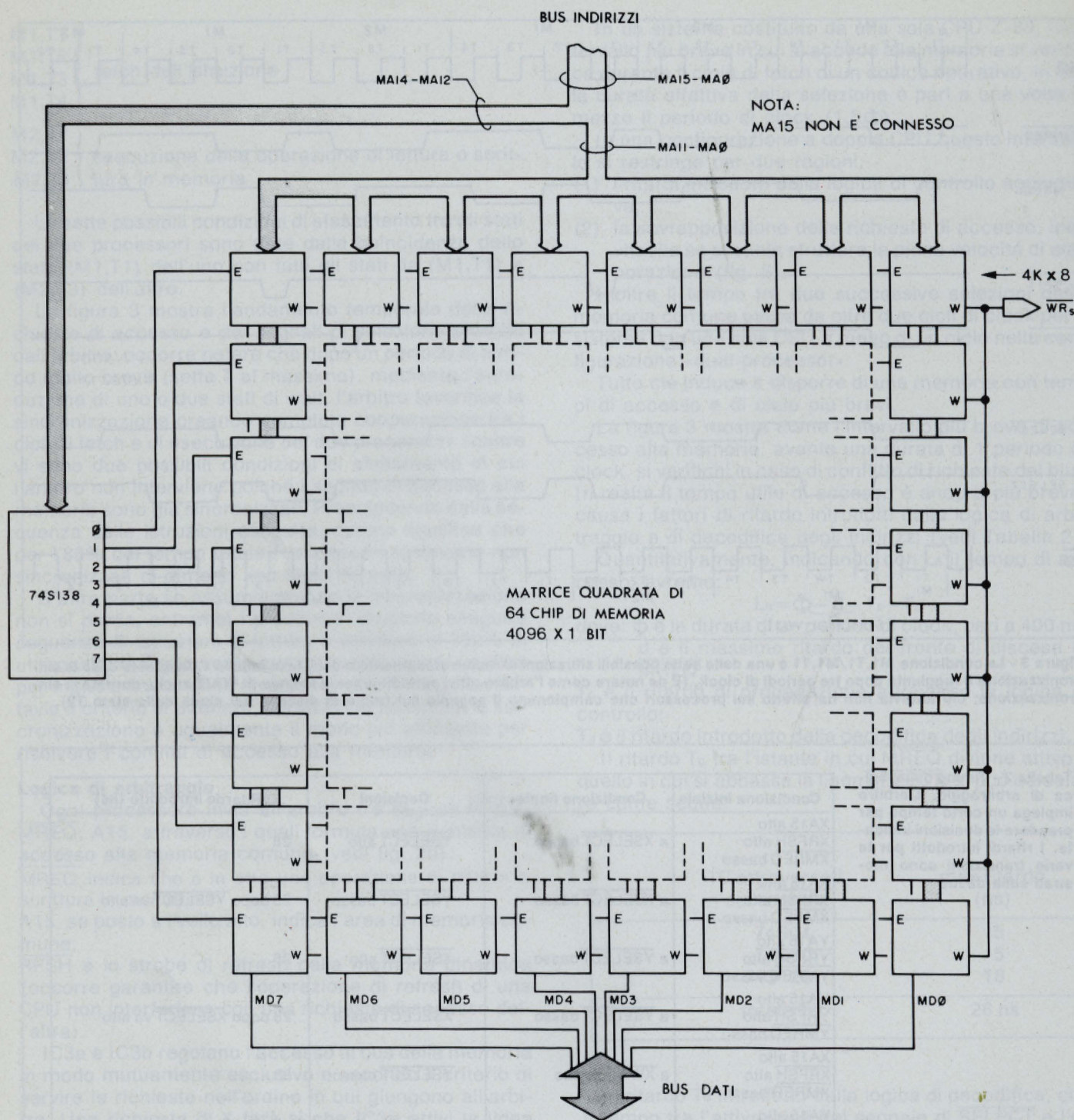



Figura 4 - Schema a blocchi della memoria condivisa. E' costituita da una matrice quadrata di 64 chip di RAM statica, ognuno di 4096 x 1 bit.

no, ad alta velocità, diverse funzioni tra cui lo smistamento dei messaggi alla stazione destinataria, il controllo di ridondanza, eventuali formattazioni, ecc.

Chiaramente un microprocessore non è in grado di svolgere queste funzioni nei limiti di tempo imposti dalla velocità di trasmissione, per cui si deve spesso ricorrere a sistemi più complessi e sofisticati. L'alternativa a questa scelta può essere un sistema multiprocessor in cui tutte le operazioni sopra indicate sono elaborate in parallelo, con una notevole riduzione di costi.

Per tutti coloro che operano con sistemi dedicati, una configurazione multiprocessor non è più di una interessante applicazione circuitale; solo nella necessità di elevare la capacità eleborativa di un sistema si può infatti apprezzare questa soluzione.

Un'altra possibile applicazione potrebbe essere l'aggiunta di un microprocessore più potente ad un sistema dotato di un'altra CPU, allo scopo di elevarne le prestazioni, ottenendo una configurazione «dual processor».

Un'area di memoria comune potrebbe rendere particolarmente semplice il problema dell'interfacciamento della nuova CPU senza richiedere la configurazione dell'intero sistema. Naturalmente l'aggiunta di un secondo processore comporta la scrittura di un sistema operativo, problema questo oggi particolarmente studiato nel mondo dei microprocessori.

MM57109: Un microprocessore per applicazioni numeriche

di Peter Nelson

È possibile dotare un sistema a microprocessore di capacità aritmetiche notevoli quali calcolo di funzioni trigonometriche e logaritmiche, utilizzando un processore dedicato quale è l'MM57109 della National Semiconductor. In questo articolo è mostrato come interfacciare queste componente con un sistema 8080.

Il microprocessore MM57109, prodotto dalla National Semiconductor, è stato specificatamente studiato per applicazioni numeriche. È dotato infatti di un set di istruzioni particolarmente sofisticato, comprendente l'aritmetica floating point, le funzioni logaritmiche e trigonometriche ed altri tipi di operazioni.

Sebbene possa essere utilizzato da solo, opportunamente fornito di memoria, o come nucleo centrale di piccoli strumenti, l'applicazione più interessante è forse il suo impiego come processore periferico, asservito ad un altro computer, in grado di assolvere al calcolo numerico.

Il componente richiede una tensione di alimentazione di 9 volt, che può essere suddivisa in + 5 e − 4 volt, per consentire un più facile interfacciamento con logica TTL; necessita, come base dei tempi, di un clock a fase singola avente una ampiezza di 5 volt e frequenza 400 kHz. Quest'ultimo è diviso internamente per ottenere un segnale a 100 Hz, (segnale SYNC) che rappresenta il periodo base del processore, detto microciclo, di durata di 10 µs. Nel seguito in tempi di esecuzione delle istruzioni sono riportati in microcicli.

La figura 1 mostra il lay-out dei piedini e uno schema a blocchi del componente: istruzioni e dati sono inviati al processore attraverso le linee I1 - I6 con la temporizzazione fissata dal segnare RDY. Questo rimane alto, indicando una richiesta di input, per un periodo di 8 microcicli, durante il quale un codice istruzione o un dato deve essere caricato sul bus. Il fronte di discesa di RDY causa la lettura del bus e la prosecuzione dell'elaborazione.

I comandi che prevedono l'invio di un doppio codice

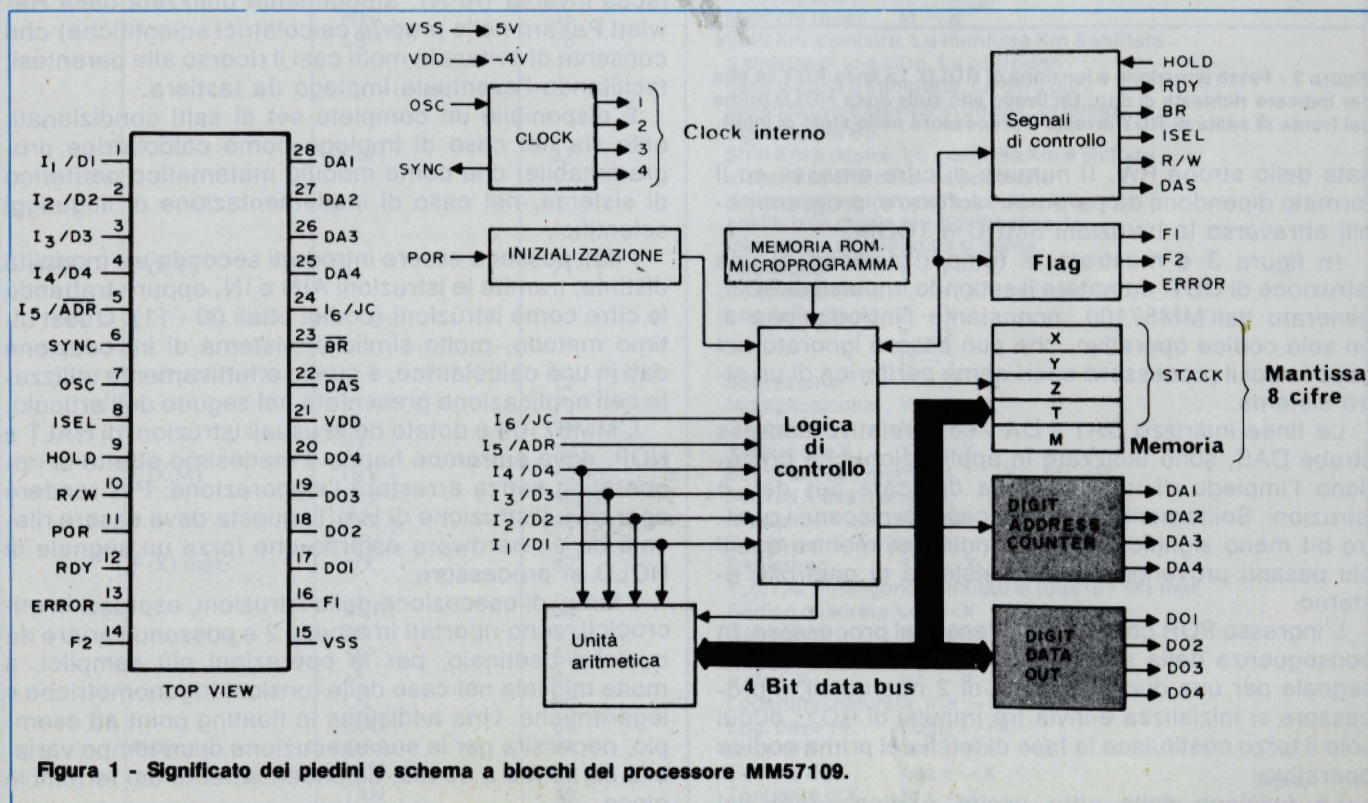



Figura 1 - Significato dei piedini e schema a blocchi del processore MM57109.

operativo causano, dopo l'introduzione della prima parola, la generazione di un altro impulso RDY, per il caricamento della seconda.

Vi è la possibilità di allungare indefinitamente il ciclo di input, applicando un livello alto all'ingresso HOLD prima del fronte di salita di RDY.

La funzione del segnale di HOLD, mostrata in figura 2, consente di rallentare l'elaborazione del processore, concedendo più tempo alla logica esterna per fornire dati o istruzioni.

Il segnale ISEL è usato, durante un ciclo di input, per indicare se il processore è in attesa di una istruzione (ISEL alto) o di un dato (ISEL basso), consentendo applicazioni in cui dati ed istruzioni provengono da due sorgenti diverse.

L'esecuzione di una istruzione OUT determina l'output delle cifre costituenti il risultato di una elaborazione sulle linee DO1 - DO4, con una temporizzazione fissata dello strobe RW. Il numero di cifre emesse ed il formato dipendono da parametri software, programmabili attraverso le istruzioni SMDC e TOGM.

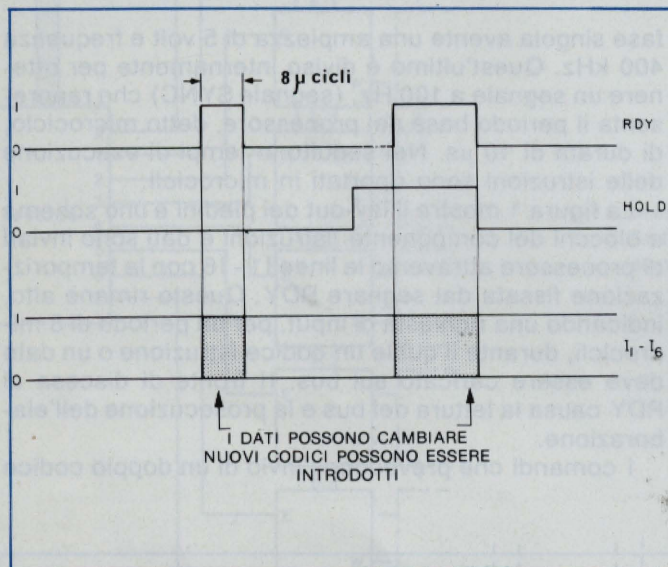



Figura 2 - Fetch istruzione e funzione di HOLD. La linea RDY va alta per indicare richiesta di dati. Un livello alto sulla linea HOLD prima del fronte di salita di RDY arresta il processore nello stato di input.

In figura 3 è mostrata la temporizzazione di una istruzione di OUT: da notare il secondo impulso di RDY, generato dall'MM57109, nonostante l'introduzione di un solo codice operativo, che può essere ignorato nel caso in cui il processore operi come periferica di un altro sistema.

Le linee indirizzo DA1 - DA4 ed il relativo address strobe DAS, sono utilizzate in applicazioni che prevedono l'impiego di una memoria dedicata per dati o istruzioni. Solitamente in questo caso forniscono i quattro bit meno significativi di un indirizzo, mentre quelli più pesanti provengono da un sistema di controllo esterno.

L'ingresso POR costituisce il reset del processore: in conseguenza della attivazione a livello alto di questo segnale per una durata minima di 2 microcicli, il processore si inizializza e invia tre impulsi di RDY, di cui solo il terzo costituisce la fase di fetch del primo codice operativo.

La funzione delle altre uscite è dipendente dal software: F1 ed F2 sono due flag che possono essere settati e resettati dalle istruzioni SF o attivati come impulsi dalle istruzioni PF; la linea ERROR segnala che è stata richiesta una operazione illegale o si è verificata una condizione di overflow; $\overline{BR}$ genera un impulso durante l'esecuzione di una istruzione di branch su condizione vera, ossia quando il salto viene effettivamente eseguito.

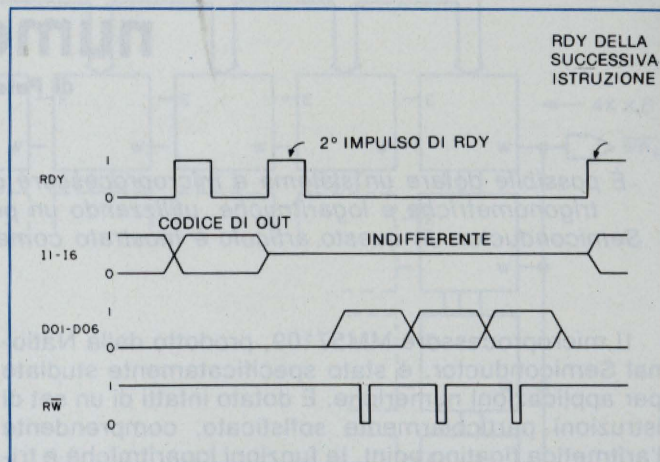



Figura 3 - Temporizzazione di una istruzione di OUT. Il secondo impulso di RDY può essere ignorato se il processore è usato come periferica di un altro computer.

Set di istruzioni

La tabella 1 elenca in dettaglio le 70 istruzioni dell'MM57109, ottenute mediante soli 6 bit di codice operativo e l'impiego della funzione INV (40 in ottale), che raddoppia le funzioni associate agli altri codici.

Il set di istruzioni comprende una serie di funzioni ricorrenti nel calcolo scientifico, e opera in notazione polacca inversa (RPN), ampiamente utilizzata dalla Hewlett Pakard nelle proprie calcolatrici scientifiche) che consente di evitare in molti casi il ricorso alle parentesi, facilitando l'eventuale impiego da tastiera.

È disponibile un completo set di salti condizionati, utile sia nel caso di impiego come calcolatrice programmabile, che come modulo matematico periferico di sistema, nel caso di implementazione di linguaggi scientifici.

I dati possono essere introdotti secondo tre modalità distinte: tramite le istruzioni AIN e IN, oppure trattando le cifre come istruzioni (codici ottali 00 - 11). Quest'ultimo metodo, molto simile al sistema di introduzione dati in una calcolatrice, è quello effettivamente utilizzato nell'applicazione presentata nel seguito dell'articolo.

L'MM57109 è dotato delle usuali istruzioni di HALT e NOP, dove entrambe hanno il medesimo effetto di nooperation senza arrestare l'elaborazione. Per rendere operativa l'istruzione di HALT, questa deve essere rilevata da un hardware esterno che forza un segnale di HOLD al processore.

I tempi di esecuzione delle istruzioni, espressi in microcicli, sono riportati in tabella 2 e possono variare da qualche centinaio, per le operazioni più semplici, a molte migliaia nel caso delle funzioni trigonometriche e logaritmiche. Una addizione in floating point ad esempio, necessita per la sua esecuzione di un tempo variabile da 22 ms a 66 ms, dipendentemente dai termini in gioco.

Tabella 1 — Istruzioni dell'MM57109. Si possono dividere in sette classi: introduzione cifre (digit entry), spostamento dati (move), funzioni matematiche (math), operazioni di cancellazione (clear), salti condizionati (brach), input output, operazioni di controllo.

CLASSE	SOTTOCLASSE	MNEMONICO	CODICE	DESCRIZIONE
Digit entry		0	00	0 Cifre per impostaz. numerica
		1	01	1 Introducendo la cifra d si ha:
		2	02	2 Z → T
		3	03	3 Y → Z
		4	04	4 X → Y
		5	05	5 d → X
		6	06	6
		7	07	7
		8	08	8
		9	09	9
		DP	12	Decimal point. Precede la parte fraz.
		EE	13	Exp.Entry. Precede l'esponente.
		CS	14	Change sign. Complementa il segno. Xm : mantissa Xe : esponente Complementa il segno di Xm o Xe a seconda che sia preceduto o meno dal comando EE
		PI	15	π Imposta 3.1415927 in X
		EN	41	Enter. Termina l'introduzione delle cifre e introduce il dato nello stack: Determina Z → T Y → Z X → Y
		NOP	77	No-operation.
		HALT	17	Halt. Per arrestare l'elaborazione occorre rilevare l'istruzione con un hardware esterno e forzare l'ingresso HOLD attivo. Altrimenti agisce come NOP.
Move		ROLL	43	Roll. Ruota lo stack: X → T → Z → Y → X
		POP	56	Pop. Determina: Y → X Z → Y T → Z o → T
		XEY	60	Scambia X,Y X → Y
		XEM	33	Scambia X,M. Scambia il registro X con la memoria: X ⇄ M
		MS	34	Memory store. X → M
		MR	35	Memory recall. M → X
		LSH	36	Shift Xm a sinistra. La mantissa Xm è shiftata a sinistra di una cifra. La posizione della virgola è immutata. La cifra più significativa è salvata in un link, la meno significativa è 0.
		RSH	37	Shift Xm a destra. La mantissa Xm è shiftata a destra di una cifra. La posizione della virgola è immutata. La cifra link sostituisce quella più significativa; la cifra meno significativa si perde.
Math	F(X,Y)	+	71	Addizione. X + Y → X Le operazioni +, —, ×, /, Y^X causano un pop dello stack: Z → Y T → Z O → T
		—	72	Sottrazione. Y — X → X
		×	73	Moltiplicazione. Y × X → X
		/	74	Divisione. Y / X → X
		Y^X	70	Y elevato a X. Y^X → X
	F(X,M)	INV +	40,71	Somma X alla memoria. X + M → M
		INV —	40,72	Sottrae X alla memoria. M — X → M
		INV ×	40,73	Moltiplica X alla memoria. M × X → M
		INV /	40,74	Divide per X la memoria. M / X → M
	F(X) mat.	1/X	67	Reciproco. 1/X → X Y,Z,T,M rimangono immutati in tutte le F(X) mat.
		SQRT	64	Radice quadrata. $\sqrt{X}$ → X
		SQ^	63	Quadrato. X^2 → X
		10^x	62	10 elev. X 10^X → X
		e^x	61	e elev. X e^X → X
		LN	65	Logaritmo naturale. ln x → X
		LOG	66	Log. base 10. Log_{10} x → X
	F(X)trig.	SIN	44	Seno di X. sin. x → X
		COS	45	Coseno di X. cos x → X
		TAN	46	Tangente di X. tg x → X

(segue)

continua da pagina precedente

CLASSE	SOTTOCLASSE	MNEMONICO	CODICE	DESCRIZIONE
		INV,SIN	40,44	Arco seno. Arc. sin x → X
		INV,COS	40,45	Arco coseno. Arc cos x → X
		INV,TAN	40,46	Arco tangente. Arc tg x → X
		DTR	55	Converte X da gradi a radianti.
		RTD	54	Converte X da radianti a gradi.
Clear		MCLR	57	Master clear. Cancella tutti i registri e la memoria; inizializza i segnali di I/O; Pone MDC = 8,MODE = floating point.
		ECLR	53	Error. clear. 0 → Error flag.
Branch	Test	JMP	26	Salto incondizionato all'indirizzo specificato dalla seconda parola. Tutte le istruzioni di branch prevedono la seconda parola come indirizzo.
		TJC	20	Test jump. Salta all'indirizzo specificato dalla seconda parola se JC (16) è vero (=1).
		TERR	24	Test error. Salta all'indirizzo specificato dalla seconda parola se il flag error è vero.
		TX=0.	21	Test su X=0. Salta all'indirizzo specificato se X=0.
		TXF	23	Test su $\lvert X \rvert < 1$. Salta all'indirizzo specificato se $\lvert X \rvert < 1$.
		TXLTO	22	Test su X < 0. Salta all'indirizzo specificato se $\lvert X \rvert < 0$.
	Count	IBNZ	31	Incrementa la cella di memoria e salta all'indirizzo specificato se M=0.
		DBNZ	32	Decrementa la cella di memoria e salta all'indirizzo specificato se M=0.
I/O	Multi digit	IN	27	Introduzione di più cifre in X. Il processore genera 4 bit di indirizzo (DA1-DA4) e uno strobe $\overline{DAS}$. L'ingresso delle cifre è temporizzato dallo strobe. Il segnale ISEL si mantiene basso per indicare un input di dati e non di comandi. Il numero delle cifre accettate dipende dal modo di funzionamento predisposto (notazione scientifica o floating point). I dati sono caricati in X e lo stack subisce uno shift: X → Y → Z → T. Alla conclusione dell'operazione di input DA1-DA4 = 0.
		OUT	26	Emissione del contenuto di X. L'indirizzamento è identico a quello per l'istruzione IN. Ogni cifra è posta sulle linee DO1-DO4, seguita da un impulso attivo basso del segnale R/$\overline{W}$. Alla conclusione dell'operazione di output D01-D04 = DA1-DA4 = 0.
I/O	Single digit	AIN	16	Input asincrono. Una singola cifra è accettata dal processore. Il segnale ISEL = 0 indica un input di dato. La lettura non ha luogo se non quando ADR = 0 ($\overline{ISEL}$ = 0 seleziona ADR invece di 15), che indica un dato valido. F2 genera un impulso attivo basso per confermare la corretta lettura.
I/O	Flag	SF1	47	Set flag 1. F1 = 1
		PF1	50	Pulsa flag 1
		SF2	51	Set flag 2. F2 = 1
		PF2	52	Pulsa flag 2
		PRW1	75	Genera un impulso attivo basso, sulla linea R/$\overline{W}$, che può essere utilizzato come strobe o clock da un hardware esterno.
		PRW2	76	Identica funzione di PRW1. Da notare che gli ultimi due bit del codice di PRW1 sono 10, mentre i corrispondenti di PRW2 sono 01. Questi due bit possono essere caricati in un flip-flop utilizzando l'impulso R/$\overline{W}$.
Mode control		TOGM	42	Cambia il modo. Passa da notazione scientifica a floating point o viceversa, in relazione allo stato precedente. Il modo di funzionamento influenza solo il formato di I/O delle istruzioni IN e OUT. Internamente il processore opera sempre su otto cifre.
		SMDC	30	Posiziona il contatore delle cifre di mantissa al valore indicato dalla successiva parola (1 – 8).
		INV	40	Prima parola per i codici doppi. Determina la possibilità di doppia interpretazione per il codice che segue.

CODICE MNEMONICO DELLE ISTRUZIONI	TEMPO DI ESECUZIONE MEDIO (MICROCICLI)	TEMPO DI ESECUZIONE MAX (MICROCICLI)	CODICE MNEMONICO DELLE ISTRUZIONI	TEMPO DI ESECUZIONE MEDIO (MICROCICLI)	TEMPO DI ESECUZIONE MAX (MICROCICLI)
0-9		238	OUT		583
DP		152	IN		395
EE		151	SF1		163
CS		166	PF1		185
PI		1312	SF2		163
HALT		134	PF2		185
AIN		284	PRW1		130
TJC		208	PRW2		130
TX=0		278	SIN	56200	95900
TXLT0		197	COS	56200	95900
TXF		277	TAN	35000	97600
TERR		191	INV SIN	54000	93900
JMP		186	INV COS	54000	93900
IBNZ		2314	INV TAN	30200	92900
DBNZ		2314	LN	24800	92000
SMDC		163	LOG	30700	92600
XEM		812	EX	30800	93900
MS		839	10X	27400	96500
MR		1385	+	2200	6600
LSH		168	INV+ INV	1700	5000
RSH		173	(M+, M-)		
INV		166	×	3200	22700
EN		552	INV × MX	2700	21400
TOGM		157		7800	22300
ROLL		905	INV (M÷)	7300	21100
ECLR		163	1/X	4500	22800
POP		448	YX	55400	95500
MCLR		734	SQRT	7000	30200
XEY		652	SQ	3000	21900
NOP		122	DTR, RTD	9600	41700

Tabella 2 - Tempi di esecuzione delle varie istruzioni. Il tempo di esecuzione è valutato in microcicli, pari a 10 µs ciascuno.

Istruzioni di IN/OUT Modo (a) = Notazione scientifica

DA4–DA1	IN: D4 / OUT: DO4	D3 / DO3	D2 / DO2	D1 / DO1
0	Cifra più significativa dell'esponente			
1	Cifra meno significativa dell'esponente			
2	Sm	0	0	Se
3	Non usata			
4	Cifra più significativa della mantissa (la virgola segue questa cifra)			
.	.			
.	.			
MDC + 3	Cifra meno significativa della mantissa			

Istruzioni di IN/OUT. Modo (b) = floating point

DA4–DA1	DP POS	IN: D4 / OUT: DO4	D3 / DO3	D2 / DO2	D1 / DO1
2		Sm	0	0	0
3		Posizione della virgola			
4	11	Cifra più significativa della mantissa = 0 ÷ 9			
5	10	.			
.	.	.			
.	.	.			
MDC + 3	12 - MDC	Cifra meno significativa della mantissa = 0 ÷ 9			

Tabella 3 - Il formato di output in funzionamento floating point o in notazione scientifica. MDC è il contatore delle cifre di mantissa, inizializzato a 8 e modificabile dall'istruzione SMDC. Sm è il segno della mantissa: 0 positivo, 1 negativo. Se è il segno dell'esponente. DPOS è l'indicatore della posizione della virgola che può assumere un valore variabile tra 11 e 12 - MDC. Indica la cifra immediatamente a sinistra della virgola.

La figura 4 mostra la semplicità di interfacciamento dell'MM57109 come modulo periferico di un sistema.

La maggior parte dei segnali sono TTL compatibili, con eccezione degli ingressi POR e HOLD, che operano ai livelli −4 e +5 volt, per i quali è necessario impiegare un doppio comparatore tipo LM339.

Le uscite $\overline{BR}$ e $\overline{RW}$ non sono in grado di pilotare direttamente un carico TTL, per cui si rende necessario l'impiego di un inverter e di un flip-flop CMOS, che hanno la duplice funzione di offrire un carico trascurabile e di memorizzare gli impulsi prodotti su queste linee.

Le linee ERROR, RDY, DO1 - DO4, nel caso operino su carichi TTL bipolari, necessitano di resistenza di pull-down del valore di 2.2 K, connesse alla tensione Vdd (−4 V), mentre nel caso di impiego su carichi CMOS ad alta impedenza i resistori devono assumere il valore di 15 K e devono essere connessi a massa.

Occorre notare come, in questa applicazione, vi siano un certo numero di pin non utilizzati: ciò accade poichè le funzioni ad essi associate sono svolte dal processore di governo.

Il listing allegato riporta un semplice programma che consente di acquisire familiarità con l'MM57109, nel caso questo operi come periferica di un sistema 8080.

Oltre a diverse routine di utility, che possono essere impiegate in programmi applicativi più vasti, il programma consente di operare in single-step, inviando al processore numerico una istruzione alla volta.

Dopo che il carattere « : » è stato visualizzato sul display di sistema, è possibile impostare una coppia di cifre ottali corrispondente al codice della istruzione da eseguire. Se questa prevede un doppio codice, è nuovamente visualizzato il carattere « : » al quale deve seguire l'impostazione della seconda parola.

Se si imposta una istruzione di OUT (codice ottale 26), il computer passa il controllo ad una routine che legge il dato da un buffer e lo visualizza su diplay. In **tabella 3** è mostrato l'effetto prodotto sul formato del display dalle differenti istruzioni.

Prima di richiedere l'introduzione di un codice operativo viene controllato lo stato delle linee ERROR e $\overline{BR}$ ed eventualmente visualizzato rispettivamente un carattere E o B, se una di queste è rilevata attiva. Uno stato di errore è resettato solo impostando l'istruzione ECLR (codice ottale 53), mentre la linea $\overline{BR}$ è resettata automaticamente dal programma all'introduzione di un nuovo comando.

Se il programma è attivato all'indirizzo di INIT, l'8080 inizializza il processore generando un impulso POR; sono ignorati i due successivi segnali di RDY mentre sul terzo è arrestata l'elaborazione portando a livello alto la linea HOLD. Questa è mantenuta attiva per un certo tempo anche dopo il caricamento di un codice istruzione sulla porta di uscita, affinchè lo stato delle linee I1 - I4 sia completamente stabile quando riprende l'elaborazione.

Il computer rileva quando è il momento di fornire una nuova istruzione, controllando lo stato della linea RDY da programma, disponibile sulla porta di input. Alternativamente il segnale di RDY potrebbe essere connesso alla rete di interrupt del sistema, non rendendo più necessario il continuo polling della linea RDY durante l'esecuzione di un'istruzione.

Durante l'esecuzione di una istruzione di OUT, è mantenuto sotto controllo lo stato del segnale DIGOUT. Se attivo questo indica infatti che una cifra del dato in output è presente sulle uscite DO1 - DO4 e deve essere letta dal sistema. A lettura eseguita segue un reset della linea DIGOUT e l'attesa di una nuova cifra, mentre il termine dell'istruzione è segnalato al solito dall'impulso RDY.

Da ultimo occorre notare che il programma è scritto per un sistema con una configurazione memory-mapped dell'input-output: cioè le porte di ingresso-uscita sono trattate come celle di memoria. Alternativamente, per l'impiego in un sistema che utilizza un indirizzamento separato dell'I/O, occorre sostituire le istruzioni LD e STA rispettivamente con IN e OUT.

INTEGRATO	TIPO	+5V	GND	4V
IC1	74C04	14	7	
IC2	CD4013	14	7	
IC3	LM339	3		12
IC4	MM57109	15		21
IC5	LM340T-5.0			

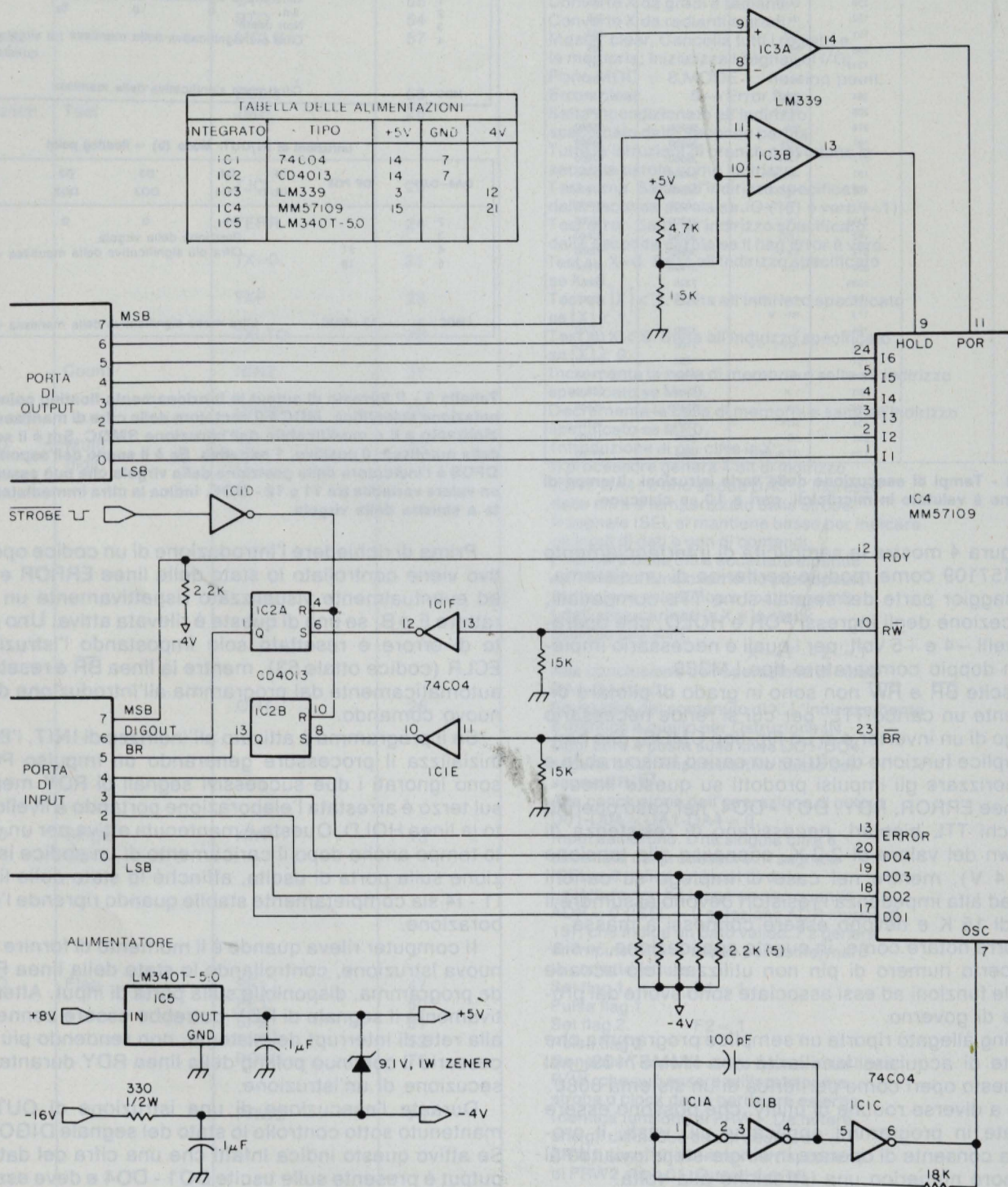



Figura 4 — Interfacciamento del processore numerico con un altro computer. Il segnale $\overline{\text{STROBE}}$ può essere controllato via software, inviando un impulso di durata compresa tra 200 ns e 50 μs. I fronti di salita e discesa non sono critici. Tutti gli ingressi non utilizzati devono essere posti a massa per evitare problemi di floating. Tutti i resistori sono da 1/4 watt a meno di altra specifica.

LISTING DEL SOFTWARE 8080 PER IL DRIVE ED IL GOVERNO DELL'MM57109

```
INIT     MVI A, BF          ;POR = 1; HOLD = 0
         STA OUT
         MVI C, 04
LOOP     DCRC               ;DURATA DELL'IMPULSO DI POR
         JNZ LOOP
         MOV A, C
         STA OUT            ;HOLD = 0; POR = 0
         CALL NXTRDY        ;ATTESA PER L'IMPULSO DI RDY
         CALL NXTRDY        ;ATTESA PER L'IMPULSO DI RDY
NEXT     ORI 40             ;PONI HOLD = 1 IN MODO CHE 57109 VADA IN
         STA OUT            ;HALT AL PROSSIMO RDY
         CALL NXTRDY        ;RIENTRO COL PROCESSORE 57109 IN HALT
NXT +    LDA INPT           ;LEGGI BR, ERROR
         RAL
         RAL
         RAL                ;BR IN CARRY
         STA FLAG           ;CLEAR FLIP-FLOP
         MOV B, A           ;SALVA ACCUMULATORE
         JNC ERROR          ;SE NON C'È SALTO
         MVI A, 42          ;IL CARATTERE ASCII B
         CALL OUTCHR        ;È VISUALIZZATO
ERROR    MOV A, B           ;RICHIAMA ACCUMULATORE
         RAL                ;ERROR IN CARRY
         JNC FETCH          ;SE NON C'È ERRORE SALTA
         MVI A, 45          ;IL CARATTERE ASCII E
         CALL OUTCHR        ;È VISUALIZZATO
FETCH    MVI A, 3A          ;IL CARATTERE ASCII:
         CALL OUTCHR        ;È VISUALIZZATO
         CALL OCTIN         ;PRENDI IL CODICE ISTRUZIONE
         CALL INSTRCT       ;INVIA IL CODICE AL PROCESSORE 57109
         CPI 16             ;SE IL CODICE INVIATO PREVEDE
         JZ OUTC            ;L'USCITA DATI DAL 57109 VA IN OUTC, ALTRIMENTI
         JMP NEXT           ;PROSSIMA ISTRUZIONE
;OUTC LEGGE LE CIFRE IN OUTPUT E LE VISUALIZZA SU DIPLAY
OUTC     LXI H, TOP         ;PUNTATORE BUFFER
NOTYET   LDA INPT           ;LEGGI LA PORTA DI INPUT
         RAL
         RAL                ;DIGOUT NEL CARRY
         JNC NOTYET         ;DIGOUT = 0?
NXTDIG   RAR
         RAR                ;RIPOSIZIONA LA CIFRA IN ACCUMULATORE E
         MOV M, A           ;MEMORIZZALA NEL BUFFER
         DCR L              ;DECREMENTA IL PUNTATORE
         STA FLAG           ;RESETTA I FLIP-FLOP DI DIGOUT
         MVI A, 40
         STA OUT            ;SETTA HOLD = 1 PER FERMARE IL PROCESSORE ALLA
AGAIN    LDA INPT           ;FINE DELL'ISTRUZIONE DI OUT DI 57109
         RAL                ;PORTA RDY NEL CARRY
         JC DISPLAY         ;SE RDY = 1, SALTA A DISPLAY
         RAL                ;PORTA DIGOUT NEL CARRY
         JNC AGAIN          ;DIGOUT = 0?
         JMP NXTDIG         ;MEMORIZZA LA NUOVA CIFRA
DISPLAY  MVI A, 0C          ;CALCOLA IL NUMERO DI
         SUB L              ;CIFRE
LISTING  MOV C, A           ;SALVA IL CONTATORE IN C
         LXI H, TOP         ;PUNTA INIZIO BUFFER
MORE     MOV A, M           ;PRENDI PRIMA CIFRA E DI QUESTA CONSIDERA
         ANI 0F             ;SOLO I QUATTRO BIT MENO SIGNIFICATIVI
         CPI 0A
         JC «0-9»           ;SE LA CIFRA È 0-9, SALTA; ALTRIMENTI
         ADI 07             ;CONVERTI A-F
«0-9»    ADI 30             ;CONVERTI TUTTI I DIGIT IN ASCII
         CALL OUTCHR        ;DISPLAY CARATTERE
         DCR L              ;DECREMENTA IL PUNTATORE
         DCR C              DECREMENTA IL CONTATORE
         JNZ MORE           ;SE ALTRE CIFRE DA VISUALIZZARE, SALTA A MORE
         JMP MEXT +         ;PROSSIMA ISTRUZIONE
;NXTRDY ATTENDE UN IMPULSO POSITIVO SULLA LINEA RDY E QUINDI RIENTRA
NXTRDY   LDA INPT           ;LEGGI LA PORTA DI INPUT
         RAL                ;RDY NEL CARRY
         JC NXTRDY
NOTRDY   LDA INPT
         RAL
         RC                 ;SE RDY È ALTO RITORNA
         JMP NOTRDY         ;ATTENDI ANCORA

;OCTIN RICEVE DUE CIFRE E LE CONVERTE IN UN VALORE DI 6 BIT IN ACCUMULATORE, CORRISPONDENTE A UN DOPPIO CODICE OPERATIVO
DELL'MM57109. SI SUPPONE DI UTILIZZARE UNA TASTIERA IN INTERRUPT (SEQUENZA EI, HLT, NOP), TUTTAVIA L'INPUT PUÒ ESSERE CON-
TROLLATO DA PROGRAMMA.

OCTIN             EI
                  HLT          ;ATTESA INTERRUPT TASTIERA
                  NOP
                  ANI 07       ;CONSIDERA I TRE BIT MENO PESANTI
                  RAL
                  RAL
                  RAL          ;MUOVI LA CIFRA DI TRE POSIZIONI A SINISTRA
                  MOV B, A     ;SALVA L'ACC.
                  EI
                  HLT          ;ATTESA INTERRUPT TASTIERA
                  NOP
                  ANI 07
                  ORA B        ;COMBINA LE DUE CIFRE
                  RET
;LA ROUTINE INSTRCT PONE IL CODICE ISTRUZIONE SULLE LINEE I1 - I6, LASCIANDO
;ALTA LA LINEA HOLD. SUCCESSIVAMENTE HOLD VIENE POSTO A 0
INSTRCT           ORI 40       ;AGGIUNGI HOLD = 1 AL CODICE ISTRUZIONE
                  STA OUT      ;INVIA CODICE
                  ANI BF       ;PONI HOLD = 0 SALVANDO IL CODICE ISTRUZIONE
                  STA OUT      ;INVIA CODICE
                  RET
```

LA NOTA

Quale spazio per gli "X-Computer"?

di Aldo Cavalcoli

La nuova rivista BIT, che nasce con questo numero, si presenta sulla ribalta della editoria tecnica in un momento particolarissimo per il mercato italiano dei microcomputer, o meglio di tutti quei computer che, nati attorno al miracle chip microprocessore, hanno assunto, a seconda delle loro configurazioni e ambiti di utilizzo, i nomi di Personal Computer, Home Computer, Small Business Computer, Sistema di Sviluppo, Sistema Didattico, Hobby-Kit.

Prima di individuare i limiti operativi ed anche filosofici di ciascuna denominazione, due parole sulle particolarità del mercato.

Guardando le società americane, oppure anche solo i nostri vicini Francia, Germania, Inghilterra, è possibile individuare una linea di continuità tra il chip microprocessore ed i super-super componenti X-Computer (dove X sta per Personal, Home, Small Business, ecc.).

L'utilizzo esteso dei microprocessori in campo industriale a tutti i livelli di complessità ed in settori applicativi estremamente diversificati ha portato come naturale conseguenza il crearsi, all'estero, di un mercato preparato a ricevere i vari X-Computer, soprattutto come abito mentale del cosiddetto potenziale utente.

Il mercato italiano invece muove ancora i primi passi; i microprocessori sono sì utilizzati in modo sufficientemente ampio, ma nella fase di «primo utilizzo» per un numero enorme di ditte.

Quindi è senz'altro errato parlare di accettazione e superamento di tutte le problematiche connesse con l'utilizzo industriale dei microprocessori.

Ed è proprio in questo momento che inizia l'invasione, del resto largamente prevista, degli X-Computer, in un mercato che faticosamente sta superando, con non pochi traumi di riconversione della professionalità dei tecnici, il primo stadio di impiego industriale dei microprocessori (si noti, industriale, non a livello prototipo).

E come se non bastasse, alcuni eccellenti centri italiani di know-how sull'impiego dei microprocessori, forti della esperienza fatta in questi anni, iniziano a dedicarsi alla realizzazione autonoma di X-Computer in concorrenza con il materiale importato dagli USA.

Con questa ultima considerazione non si vogliono certo biasimare queste iniziative, anzi a mio avviso sono da gratificare il più possibile: l'unico errore, del resto rimediabile, è che tali iniziative sono tardive.

Già due anni fa si parlava del microprocessore come di una materia prima che poteva essere utilizzata per realizzare, con il valore aggiunto dell'intelligenza e dell'inventiva del tecnico, prodotti senz'altro competitivi rispetto alla concorrenza degli altri paesi tecnologicamente più progrediti.

Una certa lentezza imprenditoriale ed una spesso eccessiva prudenza negli investimenti rende ora problematico il successo degli X-Computer made in Italy.

Un ottimo Small Business Computer USA è venduto in Italia per circa 3 milioni e mezzo, stesso prezzo di un equivalente prodotto nazionale con la differenza che il primo ha già una storia passata che lo rende più affidabile sotto tutti i punti di vista, unitamente ad un ampio supporto software già esistente, risultato dell'enorme possibilità di commercializzazione che il mercato americano permette.

Esistono però numerose condizioni che vanno subito prese in esame, come ad esempio le necessità, oltre che possibilità, di definire prodotti dedicati alle caratteristiche specifiche dei vari settori del mercato nazionale.

Occorrono però uno sforzo ed una fiducia imprenditoriale non più dilazionabili.

In questa rubrica della nuova rivista BIT è nostra intenzione svolgere un discorso critico e informativo sul settore Personal Computer, Home Computer, Small Business Computer che sia il più possibile stimolante per gli operatori del settore e per tutti coloro che seguono con attenzione questa branca di evoluzione del microprocessore.

La famiglia 9440 MICROFLAME (TM) e FIRE (TM) di microprocessori a 16-bit

T. A. Longo & P. W. J Verhofstadt - *Fairchild Camera and Instrument Corporation*

L'unità 9440 MICROFLAME è un microprocessore monolitico a 16-bit che, con un clock di 10 MHz, fornisce prestazioni compatibili con quelle di un minicomputer. In aggiunta alle caratteristiche di velocità, il set di istruzioni e l'architettura dell'interfaccia permettono di realizzare un microprocessore ad elevate prestazioni funzionali.

La figura 1 rappresenta la tipica configurazione di un sistema nel quale si fa uso di una CPU 9440, e dei membri complementari della famiglia FIRE.

Il cuore della struttura consiste appunto nel microprocessore 9440, che è una CPU per microcomputer a 16 bit, realizzato su di un unico chip avente la superficie di 27.000 millesimi di pollice quadrato. Costruito con la tecnologia I^3L per la logica interna, ed impiegante unità TTL Schottky a bassa potenza per le strutture Input-Output (I/0), viene presentato in un package a 40 pin DIP. Sebbene abbia differenze strutturali rispetto alla CPU facente parte della linea di minicomputer della Data General NOVA $^{a(+)}$, l'unità 9440 consente analoghe prestazioni, ed esegue il medesimo set di istruzioni. Il processore è una macchina a programma memorizzato, in cui viene impiegata una memoria esterna di tipo omogeneo, nel senso che le istruzioni ed i dati vengono immagazzinati nella medesima memoria. Sebbene il 9440 possa elaborare 16 bit di informazione, soltanto 15 bit vengono usati per l'indirizzamento della memoria. Di conseguenza, la capacità intrinseca della memoria di un sistema 9440 è di 32.768 parole di 16-bit (64K bytes). Tuttavia, la gamma di indirizzamento può essere estesa fino ad almeno 0,5 M bytes, tramite lo schema «memory mapping» del 9440.

La CPU MICROFLAME controlla l'apparecchiatura periferica di I/0, svolge le operazioni aritmetiche, logiche e di «data handling», e sequenzia il programmà. Esso consiste di un insieme di «data paths», e di tutti i necessari circuiti di controllo. La parte «data-path» comprende un banco di quattro registri di 16-bit general purpose (accumulatori AC_0 - AC_3), due multiplexer, una ALU e quattro registri speciali di 16-bit - scratch register, bus register, instruction register e program counter. I dati interni vanno ai vari registri attraverso un collegamento a 4-bit; i segnali di controllo vengono forniti ad opera di una matrice logica interna «mask programmed» (PLA). (vedi figura 2).

La logica statica del 9440 funziona con un oscillatore «ón chip», se un cristallo viene collegato tra CP e XTL, oppure può essere pilotato da un oscillatore esterno, tramite l'ingresso CP. Il circuito logico di clock presente sul chip produce tutti i segnali necessari di temporarizzazione, e funziona da d.c. a 12 MHz.

Un information bus bidirezionale, tri-state, di 16-bit, porta sia i dati, sia gli indirizzi tra la CPU ed i dispositivi periferici. I canali I/0 possono servire fino a 63 dispositivi esterni, ciascuno con tre porte bidirezionali I/0 (per un totale di 189), impiegando l'I/0 programmato, quello pilotato tramite «interrupt», o l'accesso diretto alla memoria (DMA).

Un canale I/0 viene riservato alla console dell'operatore. Sotto il controllo del programma, il 9440 può trasferire un dato di una parola di 16-bit al o dal dispositivo periferico, mentre simultaneamente controlla il funzionamento della periferica.

Il sistema di interruzione permette a qualsiasi unità periferica di interrompere il flusso normale del programma su base prioritaria. I dispositivi ad alta velocità, come le apparecchiature a nastro magnetico o a disco, possono ottenere l'accesso diretto alla memoria principale attraverso un canale dati, senza richiedere l'esecuzione di alcuna istruzione.

La prestazione del DMA è limitata soltanto dalla velocità di accesso della memoria.

(TM) *Trademark Fairchild Camera and Instrument CORPORATION*
7+8 *Trademark Data General*

I circuiti di supporto del sistema

In figura 1, intorno alla CPU 9440 sono rappresentati i diversi circuiti FIRE di supporto. Ciascuno di essi viene realizzato impiegando la tecnologia I^3L, per ottenre le prestazioni richieste dal microprocessore 9440.

Le memorie RAM Dinamiche Bipolari 93481/93483

Due memorie dinamiche I^3L ad alta densità - tipo 93481 4K x 1 e 93483 16K x 1 - sono compatibili con la CPU MICROFLAME 9440, e necessitano di una alimentazione di soli 5 V.

Gli ingressi funzionano con soglie standard TTL, e con caratteristiche termiche corrispondenti; l'uscita consiste in un «totem pole» tri-state, con una unità convenzionale di carico di 10. (vedi figura 3).

Il tempo tipico di accesso è di 100 ns, più rapido di quello delle corrispondenti memorie MOS, e la dissipazione attiva di potenza è di 400 mW, con una potenza di 70 mW in condizioni di standby. Altre prerogative sono un tempo di ciclo di 240 ns, la possibilità di scelta di due chip, il «data latching» controllabile, e tempi di ciclo e di accesso di 65 ns, quando si usa il modo «paging». Le unità 93481/93483 presentano le prerogative di alta densità e di orientamento alle applicazioni dei dispositivi di memoria bipolare, con il vantaggio economico delle RAM di tipo dinamico.

Esse vengono realizzate in unità a 16 pin DIP, con identica disposizione dei piedini per semplificare l'aumento delle dimensioni di memoria.

Controllo di memoria 9441

Il controllo di memoria 9441, realizzato in unità a 48 pin DIP, costituisce l'interfaccia tra il processore 9440 MICROFLAME ed un sistema di memoria - che può essere del tipo 93481 (4K x 1) oppure la memoria RAM dinamica bipolare 93483 (16K x 1). Esso contiene il contatore di «refresh address», ed un registro di indirizzamento di memoria di 15-bit, le cui uscite vengono multiplexate sulle linee di memory address e chip select, mediante un multiplexer da 7-bit, presente sul chip. I contatori interni producono la temporarizzazione per il segnale di address enable, write enable e row/column multiplex richiesta dalle RAM 93481/93483. Un dispositivo di sequenza interno, controllato median-

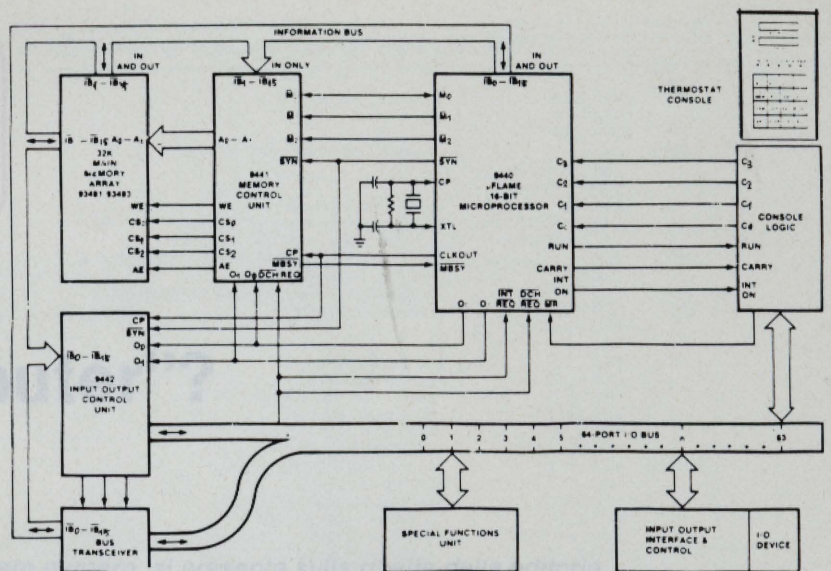



Fig. 1 — Schema del sistema basato sull'unità MICROFLAME.

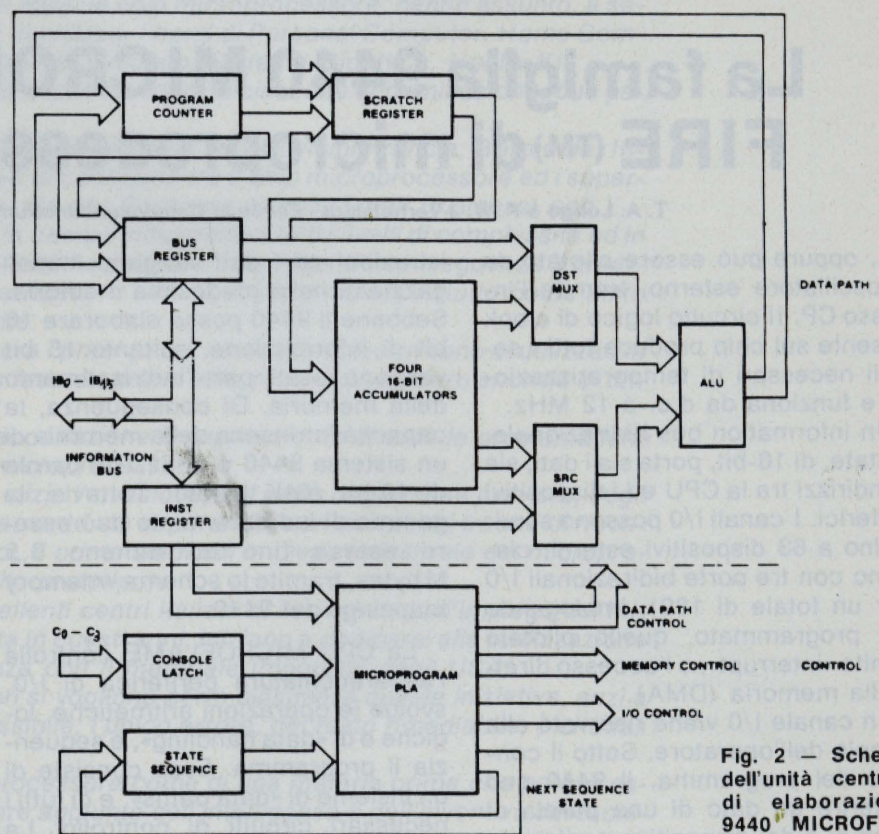



Fig. 2 — Schema dell'unità centrale di elaborazione 9440 MICROFLAME.

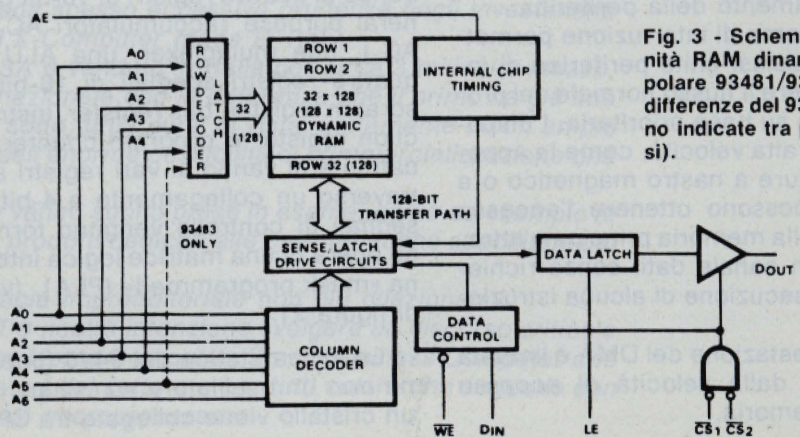



Fig. 3 - Schema dell'unità RAM dinamica bipolare 93481/93483 (le differenze del 93483 sono indicate tra parentesi).

te «logic array» controlla la temporizzazione di queste funzioni, come pure la generazione dei segnali di interfaccia del canale-dati (DMA). (Vedi figura 4).

Input/Output Bus Controller 9442

Il bus controller dell'I/0, 9442, è stato studiato per consentire l'espansione del bus di I/0 9440, per periferiche dell'utente come ad esempio le telescriventi, i terminali CRT, i sistemi di azionamento per cassette, stampanti, nastro magnetico e «Floppy - disk», nonchè per i moduli di memoria e per i dispositivi di acquisizione dati in tempo reale. Esso immagazzina e decodifica le istruzioni I/0, controlla la temporarizzazione, e contiene i «latch» per la selezione dei dispositivi e le unità pilota. I segnali di controllo vengono forniti in modo da sceglire l'input o l'output dei dati da una qualsiasi delle tre porte I/0, in un dispositivo periferico. Questi segnali, unitamente agli impulsi Start, Clear ed I/0, Pulse vengono usati per controllare un dispositivo selezionato sul bus I/0. Quando le istruzioni di controllo I/0 vengono eseguite dall'unità 9440, l'unità 9442 manda i segnali corrispondenti a tutti i dispositivi periferici, tramite i segnali di Interrupt Acknowledge, I/0 Reset, oppure Mask Out. I bit Device Select (DS_0 - DS_5) sono doppiamente bufferizzati per consentire il completamento di una sequenza di I/0, anche nel caso che venga caricata una nuova istruzione di I/0. L'I/0 bus controller 9442 può essere usato localmente, per trasformare il bus di I/0 9440 in un bus centrale di I/0, oppure, in alternativa, è possibile usare diverse unità 9442 per distribuire i dati a sistemi periferici installati a distanza. (Vedi figura 5).

Gli elaboratori Fire

Per sfruttare nel modo migliore possibile le caratteristiche particolari della famiglia FIRE, è stata sviluppata una serie di «computer board» allo scopo di soddisfare un'ampia gamma di esigenze per l'elaborazione dei dati, per l'applicazione immediata in sistemi dell'utente:

- SPARK-16, per esigenze di memorie piccole a basso costo
- FLAME-16, con prestazioni del NOVA 1200, per le esigenze di memorie economiche di maggiori dimensioni
- BLAZE-16, con prestazioni NOVA 3, per esigenze di elevate prestazioni

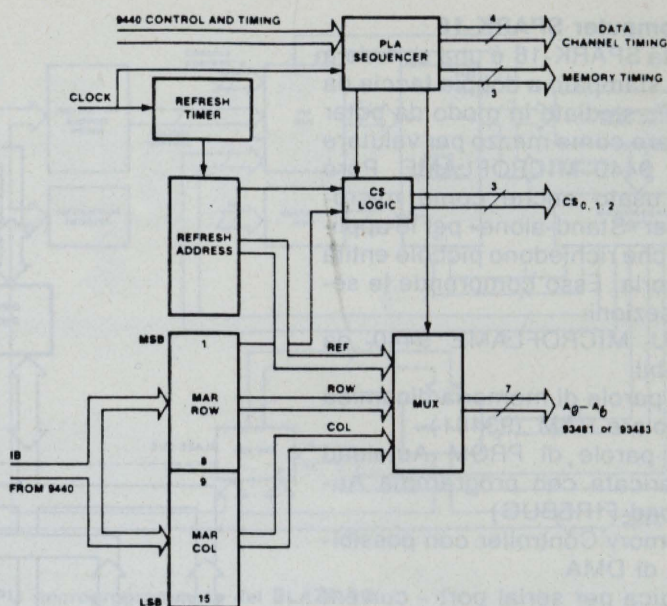



Fig. 4 — Unità di controllo della memoria 9441.

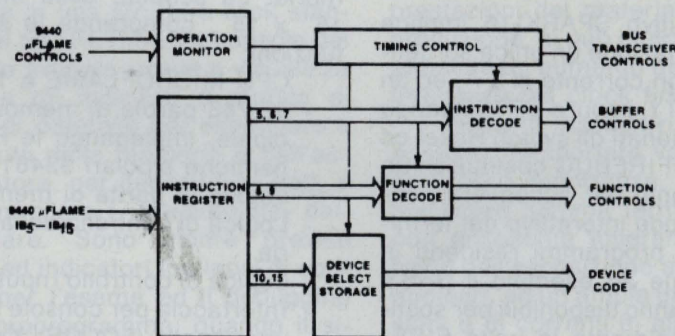



Fig. 5 — Unità di controllo del Bus di I/0 tipo 9442.

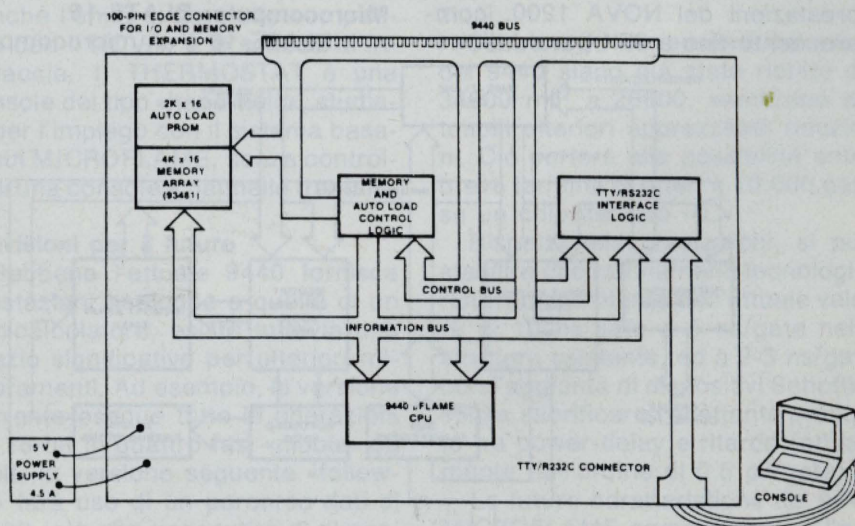



Fig. 6 — Schema della piastra del microelaboratore SPARK-16.

Microcomputer SPARK-16

L'unità SPARK-16 è una scheda a circuito stampato a doppia faccia da 8" x 10", studiato in modo da poter funzionare come mezzo per valutare la CPU 9440 MICROFLAME. Purò essere usato anche come microcomputer «Stand-alone» per le applicazioni che richiedono piccole entità di memoria. Esso comprende le seguenti sezioni:

- CPU MICROFLAME 9440 da 16-bit
- 4K parole di memoria dinamica bipolare RAM (93481)
- 2K parole di PROM Autoload (Caricata con programma Autoload FIREBUG)
- Memory Controller con possibilità di DMA
- Logica per serial port - current loop (TTY) oppure interfaccia RS232C
- Connettore laterale, con il bus 9440, a 100 pin
- Connettore per TTY/RS232C
- Switch di controllo e display (Autoload, Continue, Halt, Reset).

Il dispositivo SPARK-16 implica per l'alimentazione un'unica sorgente di 5 V, con corrente di 3 A, ed un terminale TTY oppure CRT. Quando vengono azionati gli switch Reset ed Autoload, il FIREBUG costituisce un sistema pronto per accogliere i comandi in modo interattivo dal terminale. Simili programmi residenti in PROM, come ad esempio il BABY BASIC, saranno disponibili per sostituire opzionalmente le unità FIREBUG PROM. (Vedi figura 6).

Microcomputer FLAME-16

La scheda FLAME-16 è un microcomputer bipolare da 16-bit, con le prestazioni del NOVA 1200, comprendente fino a 32K parole di memoria. La scheda, della misura di 15" x 15", comprende le seguenti funzioni:

- CPU MICROFLAME a 16-bit
- 32.768 parole di memoria principale, impiegando le RAM dinamiche bipolari 93481 4K
- Logica di parità di memoria
- Logica di controllo della memoria
- Logica di controllo Input/Output
- Interfaccia per console sia convenzionali (FOCUS XVI), sia «hand-held» (Thermostat)
- PROM per logica di AUTOLOAD e FIREBUG (2K x 16). (Vedi figura 7)

Fig. 7 — Schema della piastra del microelaboratore FLAME-16.

Microcomputer BLAZE-16

Il BLAZE-16 è un microcomputer microprogrammato da 16-bit, che viene realizzato con la serie degli elementi costituenti il microprocessore «bit-slice» Macro-Logic. Il complesso risulta compatibile con il microelaboratore MICROFLAME 9440, come pure col NOVA 3, ed esegue le medesime istruzioni. Realizzato su due schede a circuito stampato da 15" x 15", il BLAZE-16 comprende i sottosistemi qui di seguito elencati (Vedi anche figura 8).

Scheda 1 - BLAZE-16/C

- CPU microprogrammata «macrologic»
- Logica di controllo della memoria
- Sistema di memorie da 16K parole (impiegante RAM 93481 da 4K x 1)
- Logica di parità di memoria
- Logica di «Memory mapping» (indirizzamento fino a 256 K parole)
- PROM per logica di Autoload e FIREBUG
- Interfaccia del bus Input/Output

Scheda 2 - BLAZE-16M

- Un sistema di memoria da 16K parole (impiegante RAM 93481 da 4K x 1)
- Logica di interfaccia TTY/RS232C
- Logica di interfaccia per lettore/perforatore di banda ad alta velocità
- «Real time clock»

Fig. 8 — Schema del minicomputer BLAZE-16.

La CPU BLAZE-16 (Vedi figura 9) può essere sostanzialmente suddivisa in due parti principali - un percorso dati a 16-bit, ed un controllo a microprogramma. Il percorso dati consiste in quattro «Stack» di registri logici aritmetici tipo 9405A, ciascuno con otto registri/accumulatori, e quattro switch per linea dati, 9404. Un sequencer per microprogramma tipo 9408 I^3L controlla il microprogramma di 1024 parole da quarantotto bit, e risiede in dodici unità ROM da 1024 x 4. In aggiunta al set fondamentale di istruzioni FIRE, il microprogramma BLAZE-16 elabora le istruzioni di moltiplicazione e di divisione, e svolge le operazioni di «stack handling». Esso comprende una routine di auto-test, e 15 routine per la console dell'operatore.

All'incirca il 70% della memoria del microprogramma è libero per la microprogrammazione da parte dell'utente, e per la futura espansione del set di istruzioni. Il BLAZE-16 funziona in modo «pipeline», con microciclo di 200 ns. Un bus bidirezionale di 16-bit, comune a tutti i dispositivi periferici come lo è quella del sistema basato sull'unità 9440, trasferisce sia gli indirizzi che i dati tra la CPU e la memoria principale. I sistemi di memoria sono stati studiati in modo da adattarsi alle unità RAM 93483 da 16 x 1 di tipo bipolare, ciò che consente di espandere il BLAZE-16 trasformandolo in un minicomputer da 128K - parole.

Fig. 9 — La CPU microprogrammata del BLAZE-16.

Fairchild Integrated Real-time Executive (FIRE) software

Il set di istruzioni della CPU 9440 MICROFLAME risulterà familiare a coloro che hanno esperienza col set di istruzioni NOVA. Esso consiste in 2192 diverse istruzioni divise in quattro comode classi - Riferimento a Memoria, con o senza registro, Aritmetiche/Logiche, ed Input/Output.

Per realizzare i prodotti basati sul sistema MICROFLAME, in una grande varietà di applicazioni, è stato sviluppato il complesso sistema di software FIRE. Esso comprende i programmi che si estendono dal più semplice «bootstrap» e dai «binary loaders» (FIRE-LOAD) ai più sofisticati FIRE-FORTRAN e FIRE-IDOS. Inoltre, un «cross macro-assembler», un «cross linking loader» ed un «cross simulator debugger» sono disponibili su base «time-sharing» in tutto il mondo, per lo sviluppo dei programmi.

Focvs XVI e Thermostat

Il contenitore per calcolatore FOCUS XVI si presenta con uno chassis a nove guide, un alimentatore regolato ed una console a pannello frontale. È stato studiato per l'impiego «stand-alone», oppure per potersi adattare in una struttura rack standard del tipo RETMA. La console del pannello frontale riceve e visualizza i dati e gli indirizzi; è munita di switch di controllo per esaminare il contenuto della memoria e dell'accumulatore, per depositare i dati, e per controllare i «breakpoints» dell'hardware. Sono inoltre previsti switch ed indicatori luminosi di controllo, per l'esame ed il debugging del microprogramma, quando il sistema FOCVS viene usato con il BLAZE-16. Lo chassis è in grado di accogliere fino a nove schede a circuito stampato da 15" x 15", compresi il FLAME-16, il BLAZE-16, nonchè l'unità di controllo compatibile con il NOVA, e le schede di interfaccia. Il THERMOSTAT è una console del tipo «hand-held», studiato per l'impiego con il sistema basato sul MICROFLAME, senza controllo di una console a pannello frontale.

Previsioni per il futuro

Sebbene l'attuale 9440 fornisca prestazioni analoghe a quelle di un minicalcolatore, esiste tuttavia uno spazio significativo per ulteriori miglioramenti. Ad esempio, la versione corrente esegue tutte le operazioni da 16-bit in quattro fasi «nibble» da 4-bit, la versione seguente «follow-on» farà uso di un percorso dati di 16-bit, ciò che consentirà di dimezzare i tempi effettivi di istruzione. Unitamente a questo miglioramento, verrà tradotta in pratica una prestazione di moltiplicazione/divisione da hardware che migliorerà le prestazioni per le applicazioni principalmente di calcolo, come ad esempio il controllo numerico, la navigazione, ecc. In aggiunta considerando le prestazioni del materiale di attuale produzione, si può concludere che è attuabile un aumento della massima frequenza del clock fino a 15 MHz.

I moderni dispositivi del tipo I^3L possono comportare vantaggi di significativa importanza, nei confronti dell'intera gamma termica militare, con eccellente tolleranza nei confronti dell'esposizione alle radiazioni nucleari, ed un'ampia gamma di tensioni e di correnti di alimentazione, elevata capacità di pilotaggio di uscita, combinazioni digitali-analogiche, ecc. Tuttavia, esistono anche considerevoli opportunità di miglioramento di densità e prestazioni.

Sebbene le dimensioni del «die» del 9440 siano già state ridotte da 34000 mil^2 a 26600, sembrano attuabili ulteriori apprezzabili riduzioni. Ciò porterà alla possibilità entro breve termine di ottenre 10.000 gate su un chip del tipo I^3L.

Rispetto alle prestazioni, si può stabilire che raffinamenti tecnologici ridurranno il ritardo dall'attuale valore di 10 ns/gate a 5 ns/gate nella struttura esistente, ed a 2-3 ns/gate con l'aggiunta di dispositivi Schottky senza sacrificare l'allettante prodotto tra power-delay e ritardo (attualmente dell'ordine di 0,5 pJ/gate).

Le future caratteristiche del 9440 MICROFLAME comporteranno l'aggiunta di queste e di altri raffinamenti.

Nuovo HP 250.

Fino ad ora, i computer a basso costo erano anche macchine abbastanza limitate, perché nessuno riusciva a rendere semplice un Sistema potente. Noi ci siamo riusciti: un vero elaboratore gestionale per 21 milioni!

Fino ad oggi gli elaboratori gestionali a basso costo erano anche caratterizzati da un basso livello di sofisticazione. Questo perché nessuno riusciva a far funzionare un Sistema complicato e potente in maniera semplice.

Noi ci siamo riusciti. L'HP 250 è incredibilmente semplice da usare, ma è anche veloce, ha sistema operativo e gestione della base di dati per eseguire operazioni complesse sia in linea che in batch: questo significa avere a disposizione un Sistema veloce e sofisticato senza bisogno di assumere un esperto per farlo funzionare.

Un nuovo elaboratore con gestione della base di dati e altre potenti funzioni software per l'ingresso e l'uscita dei dati, con memoria di Sistema fino a 192 Kbytes e memoria

La semplicità.

d'utente di 64 Kbytes, possibilità di auto-test, e prezzo che parte da 21 milioni.

Il nostro elegante computer è anche semplice da programmare: con il BASIC gestionale HP, l'IMAGE/250 (il software DBM), il QUERY/250, il FORMS/250 e il REPORT WRITER/250 hai a disposizione tutto per ottenere dal tuo Sistema il meglio in meno tempo.

Incorporati nello schermo regolabile ci sono otto tasti software estremamente utili: il significato di ognuno viene visualizzato sullo schermo, consentendo all'operatore di far girare un programma o di

inserirne uno nuovo in un istante.

Sia la memoria centrale che l'unità di elaborazione ad alta velocità sono sotto il piano della scrivania: e basta far scorrere lo sportello sul fianco per avere accesso all'unità a dischi flessibili (ne puoi usare fino a tre, o aggiungere dischi a cartuccia per aumentare la capacità di memorizzazione).

La tastiera dell'HP 250 è identica a quella di una macchina da scrivere, con la sezione numerica a 10 tasti uguale a quella delle addizionatrici: chiunque si sieda di fronte a questo elaboratore si sentirà immediatamente a suo agio.

Per i tabulati, puoi scegliere tra due modelli di stampanti.

E il modulo fonoassorbente opzionale è cosí effecace che puoi installare il Sistema perfino nell'ufficio del presidente.

Detto questo, devi convenire che il prezzo dell'HP 250 è una piacevole sorpresa.

Non vale la pena di telefonarci o di scriverci subito per saperne di piú?

Italia: Via Di Vittorio, 9 - 20063 Cernusco sul Naviglio (MI)
Tel. 903691 - Altri uff.: Roma, Torino, Padova, Bologna, Napoli

COLT-8: Sistema di sviluppo universale o personal computer?

a cura della Microlem S.p.A.

Il microcomputer COLT-8 è un sistema molto versatile e potente. Ha la possibilità di utilizzare come scheda di CPU una scheda differente a seconda del tipo di microprocessore che si intende usare.
La memoria può essere espansa fino ad un massimo di 64K bytes; inoltre è dotato di altre tre schede per l'interfacciamento con le periferiche standard di cui è fornito, e che sono tastiera, video, cassetta magnetica e minifloppy.
Altro fatto peculiare è che il bus di sistema è un bus standard, e cioè il bus S-100.

PERIFERICHE

Tastiera È di tipo professionale e rappresenta la principale interfaccia tra il sistema COLT-8 e l'utente. Presenta il set grafico completo codificato in ASCII. La trasmissione dei caratteri è seriale con interfaccia RS-232.

Cassetta magnetica È di tipo digitale con modo di registrazione NRZ. Anche questa periferica si interfaccia serialmente col sistema in modo RS-232.

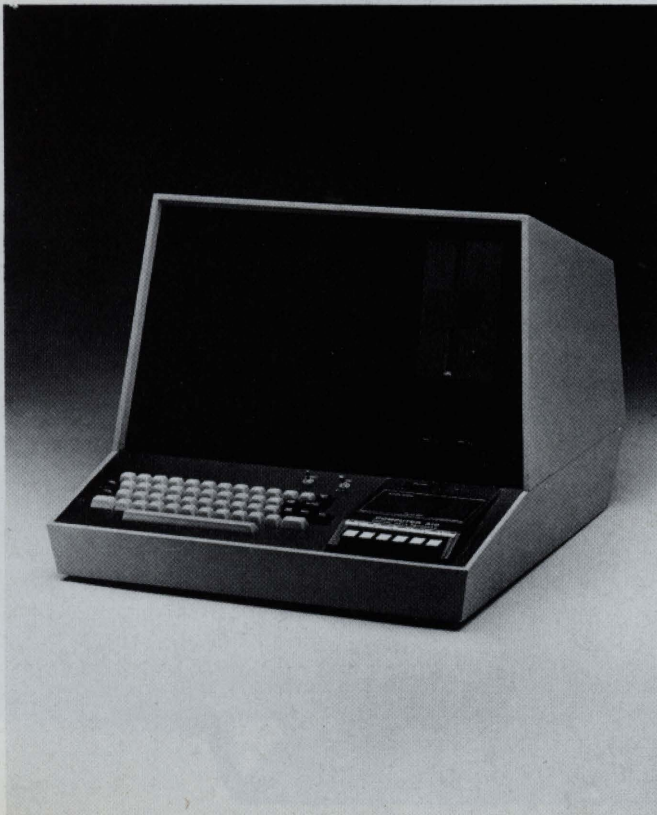
Minifloppy disk Questà unità utilizza il drive Shugart ed è supportata da una completa logica di controllo sviluppata su di una scheda a parte (scheda VERSAFLOPPY) che rende il colloquio con la scheda di CPU molto semplice.

Il dischetto minifloppy è protetto da errate manovre di accesso allo sportello mediante due sensori. Vi è inoltre l'opzione di attivare il motore del drive solo in fase di indirizzamento della periferica (invece che alla accensione del sistema), allo scopo di allungare la vita del supporto magnetico.

Video È una unità CRT della Hitachi gestita da una apposita scheda (Interfaccia video) che in particolare genera i segnali di sincronismo e quelli di video necessari per il pilotaggio del video. Questi segnali (video output, sincronismo x, sincronicmo y) sono accessibili dal connettore della scheda di interfaccia video.

HARDWARE

Scheda CPU Come è stato accennato, il sistema COLT-8 può utilizzare varie schede di CPU. In particolare le schede sviluppate sono relative all'8080, 8085, Z-80 e 6800, tutte con pin-out per il bus S-100.

Ciascuna scheda contiene, oltre alla logica necessaria alle varie funzioni di sistema (clock, logica di indirizzamento, DMA, interrupt, segnali di controllo, etc.), anche delle EPROM con il programma di monitor.

Scheda memoria Questa scheda può contenere fino ad un massimo di 64K bytes di memoria RAM dinamica. È inoltre dotata della logica necessaria per il refresh.

La memoria può essere parzialmente disabilitata in modo dinamico da un segnale (PHANTOM DISABLE) allo scopo di poter vedere sempre nella loro totalità i 64K bytes di memoria (meno un numero variabile di locazioni di memoria comunque disposte). Questa soluzione permette di risolvere i problemi di espansione dell'INPUT/OUTPUT in maniera efficiente.

Scheda floppy-disk (VERSAFLOPPY) Questa scheda implementa la logica di comando del drive floppy-disk.

È stata sviluppata attorno al chip di controller-formatter FD 1771B-1 della Western Digital, il che permette, tra l'altro, di utilizzare soltanto 4 indirizzi di I/O e di poter pilotare, mettendo i drives opportuni, sia mini-floppy a semplice/doppia densità e a singola/doppia faccia, sia floppy a semplice densità e a singola/doppia faccia, questi ultimi in formato compatibile IBM.

Inoltre questa scheda può gestire fino ad un massimo di 4 drives.

Scheda interfaccia video In questa scheda ha sede una RAM di refresh video di 1K bytes per cui è possibile realizzare un formato video di 64 righe e 16 colonne. È possibile sostituire i chips di memoria RAM 1024x1 con chips da 4096x1 per poter visualizzare fino a 4 pagine di memoria in tempi e modi controllati da programma.

Sono ammesse due modalità di visualizzazione: 1) grafica; 2) display enhancement.

Nella modalità grafica il contenuto della RAM di refresh è presentato direttamente al video, senza passare attraverso il generatore di caratteri, per cui i singoli bit sono interpretati come punti da visualizzare.

Nella seconda modalità il bit più pesante di ogni carattere decide se il carattere è nero su bianco («O») o bianco su nero («1»).

Queste modalità sono gestite dal programma utente. Inoltre è possibile ottenere la combinazione delle due modalità.

Il generatore di caratteri è in grado di fornire 128 caratteri rappresentanti il set alfabetico latino (maiuscolo e minuscolo), i simboli previsti dalla convenzione USA-ASCII, i caratteri ASCII di controllo (rappresentati sotto forma di lettere dell'alfabeto greco).

Scheda SIO Contiene due interfacce RS-232 o 20 mA current loop, una di tipo UART e l'altra di tipo USART.

L'UART è dedicato al controllo della tastiera e della stampante eventuale (oppure può essere utilizzato per il collegamento di un terminale esterno). L'USART è invece dedicato al controllo del registratore digitale. Un «baud rate generator» completo (da 75 a 38400 BAUD) è presente sulla scheda.

Vi sono inoltre 2K bytes di EPROM contenenti programmi di servizio che verranno descritti in seguito.

Mappa di memoria Le varie piastre che compongono il sistema COLT-8 contengono delle memorie il cui impiego viene qui di seguito spiegato:

1) La piastra CPU ha una EPROM da 1K che contiene il monitor di sistema.
2) La piastra di «video interface» ha una memoria da 1K RAM per il refresh del display.
3) La piastra SIO (Dual Serial Input/Output) ha 2K di EPROM su cui risiedono le routines di base per il mini floppy-disk, quelle per la cassetta e le routines di base per il display (cursor addressing, scroll line e display editing).

FIRMWARE

Monitor Sviluppa tutte le macro-funzioni di utilizzazione del COLT-8: LOAD, CHANGE, DISPLAY, GO, HUNT, SEARCH, etc.

Bios Sviluppa tutte le macro-funzioni di floppy-disk: READ ADDRESS, READ SECTOR, WRITE SECTOR, WRITE TRACK, etc.

Routines per la cassetta Sviluppano tutte le macro-funzioni relative al controllo del registratore, come READ, WRITE, e alcune macro-funzioni complesse come SEARCH e HUNT.

Firmware di controllo schermo È responsabile dello sviluppo di tutte le macro-funzioni di display tipo indirizzamento cursore, serall, editing, etc.

Lo spazio occupato nelle due EPROM dai programmi di Bios, di controllo video e di controllo registratore è di circa 1300 bytes; ciò significa che ci sono ben 700 bytes liberi che possono essere utilizzati per inserire altri programmi di servizio come routines matematiche in quadrupla precisione (32 bit) o routines di tipo pocket calculator a 12 digits.

SOFTWARE

Nella versione Z-80 sono disponibili:

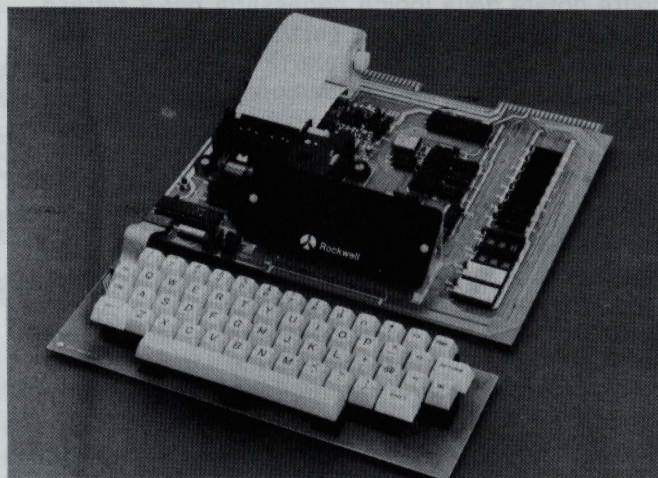
1) Sistema operativo per floppy-disk
2) Text Editor
3) Assembler
4) BASIC nelle due versione
   a) interprete
   b) compilatore
5) D BASIC versione che lavora su dischi
6) DDT: Dynamic Debugging Tool
   Effettua le funzioni essenziali di:
   BREAK POINT, INSTRUCTION STEP, MOVE CODE, RELOCATE CODE, DISASSEMBLE CODE, INSERT CODE, DELETE CODE, MODIFY CODE.
7) Programmi di utilità
   a) LINK LOADER: consente di unire vari sottoprogrammi per ottenere un unico programma oggetto eseguibile direttamente in macchina.
   b) PIP: raggruppa un insieme di funzioni quali SAVE, COPY, LOAD, LIST, operanti su files residenti su floppy-disk.
   c) Programmi di generazione o modifica del sistema operativo. Es.: espansione di un sistema operativo da 32K a 48K, oppure generazione di un sistema operativo relativo ad un altro COLT-8 con logica principale diversa da quella impiegata (6800 invece di Z-80).
   d) Debugger: equivalente dinamico del DDT chiamabile da programma. Può fare un trace pilotato e/o condizionato da programma. Es.: visualizzare il contenuto di alcuni registri solo al verificarsi di particolari condizioni o in presenza di particolari istruzioni.

A questo punto, si può senz'altro dire che con questo prodotto la differenza tra personal computer, sistema di sviluppo, sistema gestionale è veramente diventata minima.

Il privilegiare uno di questi aspetti è un fatto puramente applicativo.

AIM 65:

Sistema di sviluppo educativo a basso costo e alte prestazioni

a cura della
G. De Mico S.p.A.

Proseguendo sulla strada scelta un anno fa con la famiglia 6500 verso un mercato microcomputer più vasto e standardizzato, la Rockwell ha annunciato un sistema di sviluppo educativo completo ed espandibile a basso costo, ponendosi nella condizione di accedere a tutti i livelli di utilizzazione con uno strumento facile, potente ed economicamente poco impegnativo.

Concepito come supporto di studio, l'AIM 65 è in effetti un sistema microprocessore assemblato versatile, con una completa tastiera, un display a 20 caratteri e, unico nel suo genere, una stampa termica. Il programma monitor in firmware offre ampie funzioni di controllo e di sviluppo dei lavori.

La stampante termica a 20 colonne compresa nell'AIM 65 utilizza nastri di carte termica, con una velocità di stampa di 120 righe per minuto. Genera tutti i 64 caratteri ASCII con una matrice di stampa 5 7. È una caratteristica unica nel suo genere per un calcolatore a basso costo.

L'AIM 65 si presenta con un display di 20 caratteri reali. L'informazione è visualizzata da una striscia di caratteri monolitici a 16 segmenti. È facilmente leggibile ed, allo stesso tempo, elimina problemi di comprensione.

La tastiera dell'AIM 65 di tipo terminale libera dai limiti imposti dalle mini-tastiere esadecimali. I 54 tasti danno la possibilità di generare 70 differenti caratteri alfanumerici e funzioni operative.

Il monitor residente su memoria ROM prevede una serie di comandi di facile utilizzo (richiamabili con singole lettere dalla tastiera) per un più semplice esame dei programmi, offrendo possibilità normalmente riscontrabili solo su più complessi e più dispendiosi sistemi di sviluppo. La chiarezza del monitor non dà adito ad ambiguità: si spiega da solo quando ha bisogno di informazioni, generando anche messaggi di errore, se necessario.

Il monitor dell'AIM 65 include comandi di:

- ingresso e creazione diretta di programmi eliminando la difficoltà dell'uso del codice operativo
- listing del programma su TTY o sulla stampante
- visualizzazione e modifica dei registri e della memoria
- predisposizione di punti di arresto (breakpoints), funzioni di 'trace', revisione e correzione dei programmi in corso (debug)
- controllo della stampante termica
- trasferimento di informazioni da/a cassette o TTY
- esecuzione di programmi caricati in memorie RAM, ROM, PROM; sia residenti che aggiunte
- interfaccia con gli interpreti opzionali dei linguaggi Basic e Assembler.

Il cuore dell'AIM 65 è un'unità centrale R6502. Le provate doti di velocità e semplicità, più 65K di memoria indirizzabile, si accompagnano ad un potente set di 56 istruzioni che si avvicinano molto alle caratteristiche software di un minicomputer.

Il circuito R6532, una combinazione di RAM, ingressi/uscite, timer, è utilizzato dal monitor come memoria di servizio e per operazioni con la tastiera.

Due circuiti R6522 (versatile interface adapter) si dividono i compiti di controllare la stampante, di gestire l'interfaccia con TTY e unità a cassette, di fornire all'utilizzatore le connessioni necessarie per le applicazioni esterne al sistema; sono in tal senso disponibili 2 porte I/O di parallelismo 8, una porta seriale I/O ad 8 bit e l'accesso a due timer/contatori di 16 bit.

Due ROM da 4K X 8 R2332 contengono l'intero programma Monitor. Alcuni zoccoli di riserva permettono all'utilizzatore di espandere i programmi residenti in firmware fino a 20K byte; tali possono essere sia i programmi utente su ROM R2332 o su PROM compatibili, sia le ROM opzionali dell'AIM 65 contenenti l'Assembler o il Basic.

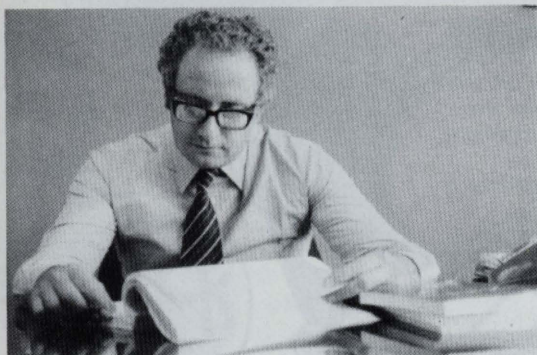
La memoria programma è fornita in due opzioni da 1 a 4K byte.

L'AIM 65 è già pronto per l'espansione

Per permettere all'AIM 65 di crescere nella maniera desiderata dall'utilizzatore, sono previsti sulla piastra un connettore per le applicazioni e un connettore per le espansioni. Il primo permette il collegamento diretto con TTY (loop di corrente 20 mA) e con una o due unità a cassette.

Ha inoltre le connessioni I/O di utilizzo generale del circuito VIA residente. Il connettore per le espansioni estende il sistema di bus dell'AIM 65 - indirizzo, dati, controlli - verso memorie addizionali esterne o comunque verso tutto ciò che vorrete collegare alla macchina.

Il linguaggio ad alto livello Basic è un'opzione residente.

TRIBUNA

Renato Baldoni
(Responsabile Marketing e Applicazioni di microprocessori e memorie - SGS-ATES)

"Crescere con il sistema"

Nell'articolo «The First Ten Years of Amateur Computing», apparso recentemente sulla rivista «BYTE», Sol Libes, presidente dell'Amateur Computer Group del New Jersey, fa risalire la nascita del «Personal Computer» al 5 maggio 1966, data di fondazione della Amateur Computer Society. Tuttavia lo stesso Libes ammette che soltanto da due o tre anni si può parlare di un vero e proprio mercato, nel senso in cui è possibile fare previsioni ed impostare scelte strategiche.

In realtà il fenomeno «Personal Computer» è esploso in questi ultimi anni in maniera incontrollata, prendendo di sorpresa anche coloro che avevano avuto il fiuto o la fortuna di entrarvi per tempo, ma non avevano saputo prevederne la portata.

Anzitutto, che cos'è un Personal Computer? Alcuni lo definiscono come «un piccolo computer autosufficiente dal costo contenuto»; definizione abbastanza squallida, ma che contiene gran parte delle caratteristiche principali.

Piccolo nel senso delle dimensioni, ma possibilmente non delle prestazioni; computer, e quindi non solo una macchina da calcolo programmabile; autosufficiente, quindi comprensivo di organi di ingresso e uscita; a costo contenuto cioè alla portata di chiunque.

Chiaramente una definizione così vaga tende a comprendere una vasta serie di prodotti, che vanno dal kit per hobbysta da poche migliaia di lire, al sofisticato computer scientifico il cui costo è di vari milioni.

Per dare un'idea del boom incredibile che si sta verificando, soprattutto negli Stati Uniti, il mercato mondiale attuale è stimato intorno ai 100 milioni di dollari, con una proiezione di 175 milioni nel 1981.

Il mercato europeo segue con notevole ritardo (proiezione di 30 milioni nel 1980) ma si prevede un forte tasso di crescita, superiore al 30 per cento annuo.

Dividere il campo dei Personal Computer in fasce applicative non è facile: si può parlare di educational computer, hobby computer, micro business computer, game computer, ma le linee di demarcazione non sono nette e l'utilizzatore è normalmente portato a cercare un sistema abbastanza flessibile da permettergli di passare da un'applicazione all'altra con minime aggiunte di hardware.

In effetti, una limitazione di alcuni sistemi esistenti è costituita dalla scarsa modularità ed espandibilità: si tratta di sistemi che risolvono molto bene un particolare problema, ma sono sistemi «chiusi» la cui espansione è impossibile o antieconomica.

Questo è particolarmente vero per la maggior parte dei sistemi «educativi» in cui, per questioni di costo, l'ingresso/ uscita è fatto in codici esadecimali.

È naturale, per chi possiede un sistema di questo genere e si è già impadronito delle tecniche di programmazione in «linguaggio macchina» il desiderio di passare a tecniche «simboliche» che richiedono organi di ingresso uscita alfanumerici completi. In un secondo tempo, utenti più sofisticati potranno desiderare periferiche ancora più potenti, come floppy-disk o stampanti seriali veloci.

È molto importante che tutte queste espansioni possano essere effettuate facilmente, senza dover praticamente riacquistare un nuovo sistema; l'utente è normalmente disposto ad effettuare un investimento iniziale un po' più alto, purchè abbia la sicurezza che il valore di questo investimento venga salvaguardato durante le successive espansioni del sistema fino al livello più alto.

Un discorso analogo si può fare per quanto riguarda la professionalità della progettazione e della realizzazione: scendere a compromessi per abbassare i costi è sempre un falso risparmio, che viene poi scontato in termini di scarsa affidabilità e prestazioni mediocri.

Sulla base di questi criteri la SGS-ATES ha progettato un sistema modulare integrato, che si presenta su tre livelli fondamentali di complessità.

Il livello minimo è chimato «nanocomputer» e consiste in un sistema educazionale molto sofisticato, dotato di un piccolo terminale con display/ tastiera esadecimale. Dato lo scopo del sistema è stata particolarmente curata la documentazione, che oltre ai manuali di uso e manutenzione si basa su un volume introduttivo in cui l'allievo viene portato passo passo dalle nozioni fondamentali di logica binaria alle più complesse tecniche di programmazione della CPU del sistema, lo Z80.

Un secondo volume è dedicato ai circuiti di interfaccia, utilizzando un bread-board disponibile come opzione.

Un kit di conversione permette di passare al livello successivo, trasformando la scheda nanocomputer nella scheda di microcomputer CLZ80, che è la base di tutte le espansioni successive. Una seconda scheda, la VDZ80 fornisce una tastiera alfanumerica completa e un'interfaccia per un televisore o un monitor televisivo.

Possiamo quindi già parlare di Personal Computer: il software disponibile su ROM permette sia la programmazione in BASIC, come la maggior parte dei Personal Computer, che quella in Assembler.

L'ingresso/uscita è realizzato da cassette magnetiche a basso costo, la cui interfaccia è standard su tutti i livelli del sistema.

Già a questo livello è disponibile una vasta gamma di opzioni, quali espansioni della memoria e dell'I/0, programmatore di EPROM, stampante seriale, eccetera.

Con l'aggiunta della scheda FLZ80, l'interfaccia per floppy-disk, il sistema raggiunge il terzo livello, che è quello di microcomputer.

Il floppy-disk dà l'accesso al disk operating system, che oltre a comprendere versioni più potenti dei programmi disponibili ai livelli più bassi, dispone anche di un sofisticato file manager e di un compilatore.

Una particolare cura è stata posta nella realizzazione di tutto un hardware di supporto (connettori, cavi, alimentatori, parti meccaniche), con cui «vestire» il sistema in maniera professionale.

Tornando ai presupposti da cui eravamo partiti, possiamo motivare le scelte della SGS-ATES con due ordini di ragioni: la prima di costi e la seconda di metodo.

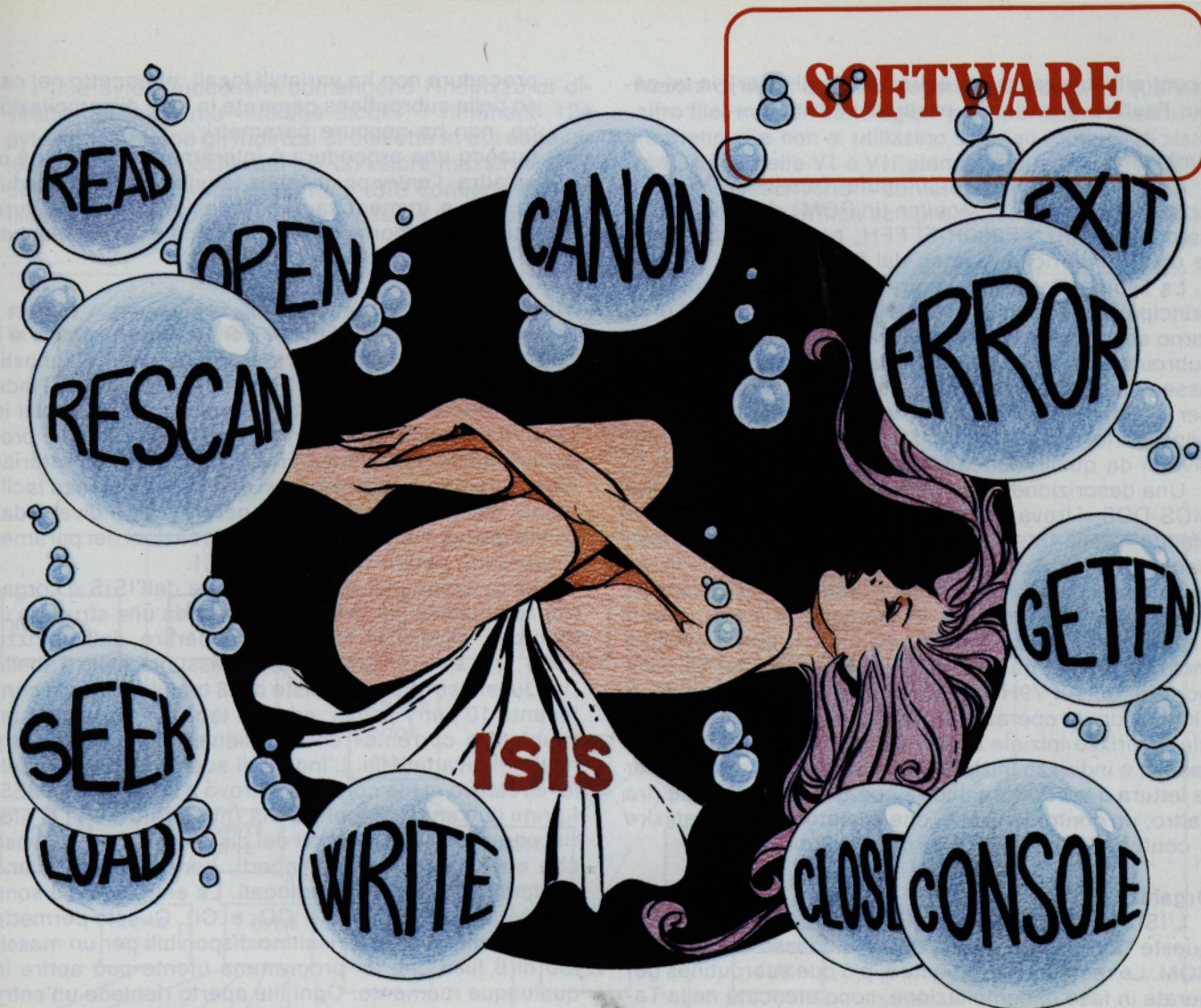
La modularità del sistema permette un approccio graduale dal nanocomputer educativo al microcomputer complesso, valorizzando di volta in volta l'investimento iniziale, ed avvicinandosi ad una professionalità sempre crescente.

Ragionando in termini di marketing ci sembra una scelta onesta, che non giustifica surrettiziamente il valore aggiunto, ma lo motiva in termini di espansione reale delle prestazioni fornite.

Per quanto riguarda il metodo, si è preferito dare all'utilizzatore la possibilità di «crescere con il sistema», ponendolo in condizioni di partire da un livello buono ma non troppo complesso e di impadronirsi di ogni gradino prima di passare al successivo.

Tale scelta si pone in netta antitesi con quella che prevede come approccio didattico l'accostamento immediato a un sistema molto complesso e programmato con linguaggi ad alto livello.

Secondo noi chi si occupa di microcomputer non può prescindere da una conoscenza approfondita della programmazione in linguaggio macchina delle tecniche di interfaccia: l'iniziare il discorso didattico con un sistema in cui questi problemi sono già risolti in maniera standard da un linguaggio ad alto livello, rischia di provocare una vera e propria «crisi di rigetto» quando l'allievo dovrà rinunciare a queste facilitazioni per motivi di costi o di efficienza.

In apparente contrasto con questa filosofia, è stato però deciso di non fornire i sistemi in versione kit. Il livello di complessità e di sofisticazione dei sistemi SGS-ATES non è, per le difficoltà di assemblaggio e soprattutto di testing, alla portata di un costruttore dilettante. È questa, o almeno ci sembra, un'ulteriore garanzia di professionalità.

ISIS: Struttura di un sistema operativo

Perry C. Hutchinson - *Computer Science Department Iowa State University*

La conoscenza della struttura interna di ISIS dell'Intel può essere utile in fase di diagnostica. È comunque stimolante in quanto permette di verificare come è organizzato un piccolo sistema operativo mono-utente.

L'ISIS, *(Intel System Implementation Supervisor)* è il sistema operativo supportato da floppy disk sviluppato dalla Intel Corporation per il microcomputer Intellec MDS. L'ISIS è stato scritto interamente in PL/M, che è un «dialetto» del PL/1 sviluppato per il microprocessore 8080. Il sistema ISIS è progettato straordinariamente bene; presenta una notevole somiglianza con il sistema operativo UNIX della Bell Labs, utilizzato per il PDP-11.

La Intel non ha mai fornito né il listing del codice sorgente, né una qualsivoglia documentazione interna dell'ISIS. Quindi, poichè una conoscenza dei processi interni di questo sistema è a volte necessaria, soprattutto quando si desidera usarlo in tutte le sue possibilità, non rimane altro che esaminare il codice oggetto. Il presente articolo presenta le principali acquisizioni di tale studio. Esso dovrebbe essere utile sia come testo di riferimento per chi ha occasione di occuparsi dei processi interni del sistema ISIS, sia come esempio didattico di quanto realmente accade in un piccolo sistema operativo monoutente. I nomi usati sono stati presi dalla documentazione ISIS pubblicata, nei casi in cui erano appropriati.

Le informazioni contenute nel presente articolo risultano acquisite dall'esame del codice oggetto dell'ISIS Versione 1,2 di 32K, che risale all'agosto 1976. (Lo stesso diskette contiene anche la Versione 1,1 dell'ASM80, la Versione 1,3 dell'EDIT, la Versione 1,1 dell'UPM e la Versione 2,0 dell'ICE80). Tranne che dove è specificamente affermata una più generale validità, non si può presumere di applicare queste informazioni a qualsiasi altra versione dell'ISIS.

È possibile che, in questa presentazione, nonostante

i controlli accuratissimi, si trovino degli errori; in tal caso, l'autore gradirebbe esserne informato.

Ambiente

L'ISIS opera sul microcomputer Intellec MDS e utilizza le prestazioni di Monitor (in ROM) dell'MDS, allocato agli indirizzi F800H-FFFFH, per comunicare con le unità periferiche diverse dal floppy disk.

La Tabella I elenca gli entry points e le funzioni delle principali routines di Monitor. I parametri e i valori di ritorno sono trattati come in PL/M. (Ci sono, beninteso, subroutines supplementari nella ROM, ma un listing di assemblaggio del Monitor è fornito col sistema MDS, per cui chi è interessato ai suoi processi interni può consultarlo. L'ISIS non utilizza entry points nel Monitor diversi da quelli elencati nella Tabella I).

Una descrizione particolareggiata del disk controller MDS-DOS si trova nel *«MDS-DOS Hardware Reference Manual»* della Intel, per cui qui ne verrà dato un breve cenno. Il controller occupa le porte di I/O 78H-7FH, ed è comandato tramite un «I/O Parameter Block» (IOPB) costruito dal programma in memoria. L'indirizzo dell'IOPB è fornito al controller inviando i relativi bytes di indirizzo (prima quello basso e poi quello alto) verso le porte di output 79H e 7AH. L'IOPB contiene le indicazioni di: drive, operazione da eseguire, indirizzo di traccia, indirizzo iniziale di settore, numero di settori interessati e indirizzo iniziale della memoria di sistema per la lettura o la scrittura. Il disk controller comprende, fra l'altro, un controller DMA, che è usato sia per acquisire il contenuto dell'IOPB, sia per trasferire dati.

Organizzazione

L'ISIS è formato da circa 57 procedure PL/M; 8 di queste forniscono puramente l'accesso al Monitor ROM. Le restanti 49 procedure, più due subroutines generate in fase di compilazione, sono elencate nella Tabella II; Le figure 1 e 2 rappresentano il legame chiamante-chiamato. (Molte delle routines della figura 1 richiamano una o più routines della figura 2, disegnate a parte per una migliore comprensione). In figura 1 le linee più scure individuano le routines chiamate direttamente dai comandi del sistema ISIS. (Le procedure ERROR e XEQIOPB sono di fatto routines di figura 2, ma per come l'ISIS le richiama sono state raffigurate in figura 1). La Tabella II dovrebbe essere ampiamente chiarificatrice di per se stessa, a parte alcune abbreviazioni e convenzioni adottate, e precisamente:

1) la seconda colonna contiene l'indirizzo di entry point della procedura;
2) le colonne contrassegnate con «T» indicano il Tipo della quantità associata, e cioè (Address, indirizzo) denota un valore di 16 bit, B (Byte) un valore di 8 bit, L (Logico) un valore di un byte usato in senso Falso-Vero. (Il valore dispari indica vero, il valore pari indica falso);
3) la colonna contrassegnata con «INDIRIZZI» indica:
A) nel caso di procedure parametriche, l'indirizzo di memoria nel quale è memorizzato il parametro;
B) nel caso di procedure non parametriche, l'indirizzo nel quale è memorizzata la prima delle variabili locali di tale procedura.
(Le variabili locali delle procedure parametriche sono memorizzate immediatamente dopo l'ultimo parametro);
4) un trattino nella colonna «INDIRIZZI» indica che la procedura non ha variabili locali, e, eccetto nel caso delle subroutines generate in fase di compilazione, non ha neppure parametri;
5) quando una procedura è interamente contenuta in un'altra, l'entry point (nella tabella) per la procedura segue immediatamente quello della procedura esterna, e il nome di quella interna è contrassegnato da due asterischi.

Strutture dei dati

Una delle principali motivazioni di questo studio è il desiderio di acquisire informazioni di utilità diagnostica. A tal fine, la Tabella III presenta un elenco di indirizzi di memoria i cui contenuti possono essere utili in caso di difficoltà; anche i valori parametrici delle procedure attive possono servire a tale scopo. La «storia» delle chiamate di procedure si ottiene abbastanza facilmente dall'esame dello stack, perchè valori diversi dagli indirizzi di ritorno sono rari. Le locazioni dei parametri possono dedursi dalla Tabella II.

L'organizzazione dei dati adottata dall'ISIS è l'organizzazione in «File» ed è controllata da una struttura di controllo situata in memoria a partire dall'indirizzo 29E5 fino all'indirizzo 2B1B, e riassunta nella Tabella IV. Questa struttura consiste di 18 tabelle, ognuna contenente 10 entry points; ad ogni tabella è associato un «puntatore corrente» che contiene l'indirizzo di entry nella relativa tabella. L'indice di scansione delle tabelle, e relativo al file corrente si trova a sua volta in 29E5. I primi due entry di ogni tabella (numerate 0 e 1) si riferiscono sempre ai direttori dei diskette 0 e 1, nel senso che questi sono sempre aperti, benchè non abbiano sempre i relativi buffers allocati. Le entrate 2 e 3 sono relative ai files di console :CO: e :CI:. Questo permette che gli entry da 4 a 9 risultino disponibili per un massimo di 6 files che un programma utente può aprire in qualunque momento. Ogni file aperto richiede un'entry in ciascuna tabella della struttura di controllo, ma la maggior parte di questi entry non sono utilizzati se il file non è su diskette.

Il parametro AFT, che è richiesto da molte procedure dell'ISIS, è semplicemente un numero compreso fra 0 e 9, che identifica una serie di entry nelle tabelle della struttura di controllo.

Il parametro AFT (ritornato alla fine esecuzione della OPEN) si differenzia dal parametro AFTN, fornito dall'utilizzatore, nel fatto che AFT = AFTN+2.

La dicitura «VALORE INIZIALE» che compare nella Tabella IV indica il valore associato al relativo elemento della struttura di controllo per un file di diskette che è stato appena aperto.

Layout di diskette

Nell'ISIS, i settori di diskette rientrano in uno di questi tre gruppi: settori di blocco-dati; settori di «linkage block»; settori disponibili. Questo è valido sia per i dati di distema che per i files di utente. Un file di ISIS è formato da «linckage blocks» e da blocchi-dati. I blocchi-dati contengono i bytes che costituiscono il file vero e proprio, mentre i «linckage blocks» dicono come i blocchi-dati sono collegati fra loro. Tutti gli indirizzi di diskette nell'ISIS sono memorizzati come Numero di Blocco e Numero di Traccia. Le tracce sono numerate da 0 a 76, e i blocchi di ogni traccia da 1 a 26. I primi due bytes di ogni «linkage block» contengono l'indirizzo di diskette del «linkage block» precedente (0, se è il primo).

I due bytes successivi contengono l'indirizzo di diskette del prossimo «linkage block». I rimanenti 124 bytes contengono gli indirizzi di diskette in cui sono allocati i relativi blocchi-dati per un valore massimo di 62 blocchi. Ogni diskette di formato ISIS contiene quattro files di sistema. Questi possono essere aperti per operazioni di input e quindi essere letti come un qualsiasi altro file, ma non sono agibili per operazioni di scrittura (a meno che non si utilizzino speciali routines di sistema). Le Tabelle VI e VII elencano questi particolari files e ne descrivono il contenuto.

L'ISIS non verifica sempre questi files di sistema nel

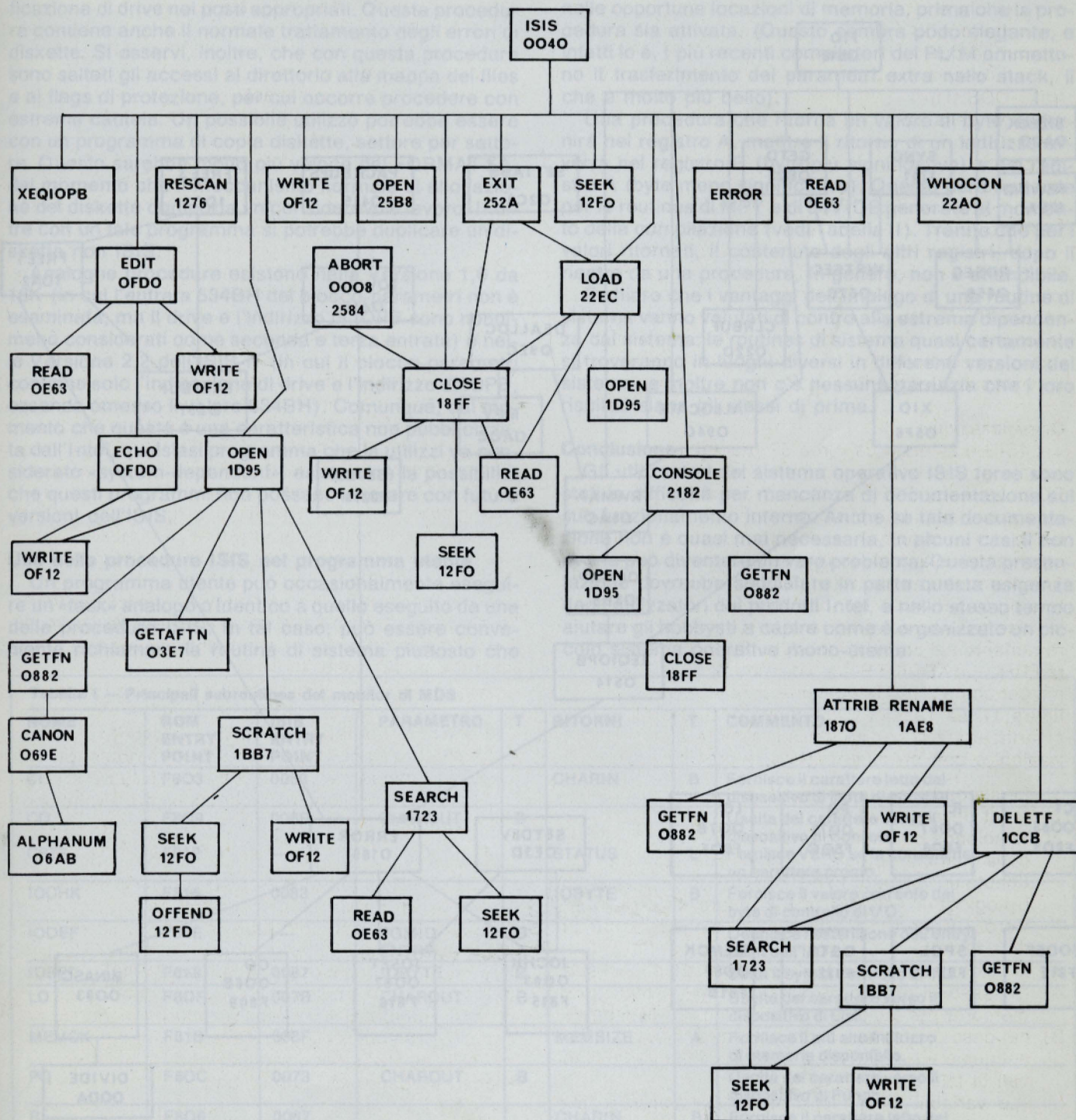
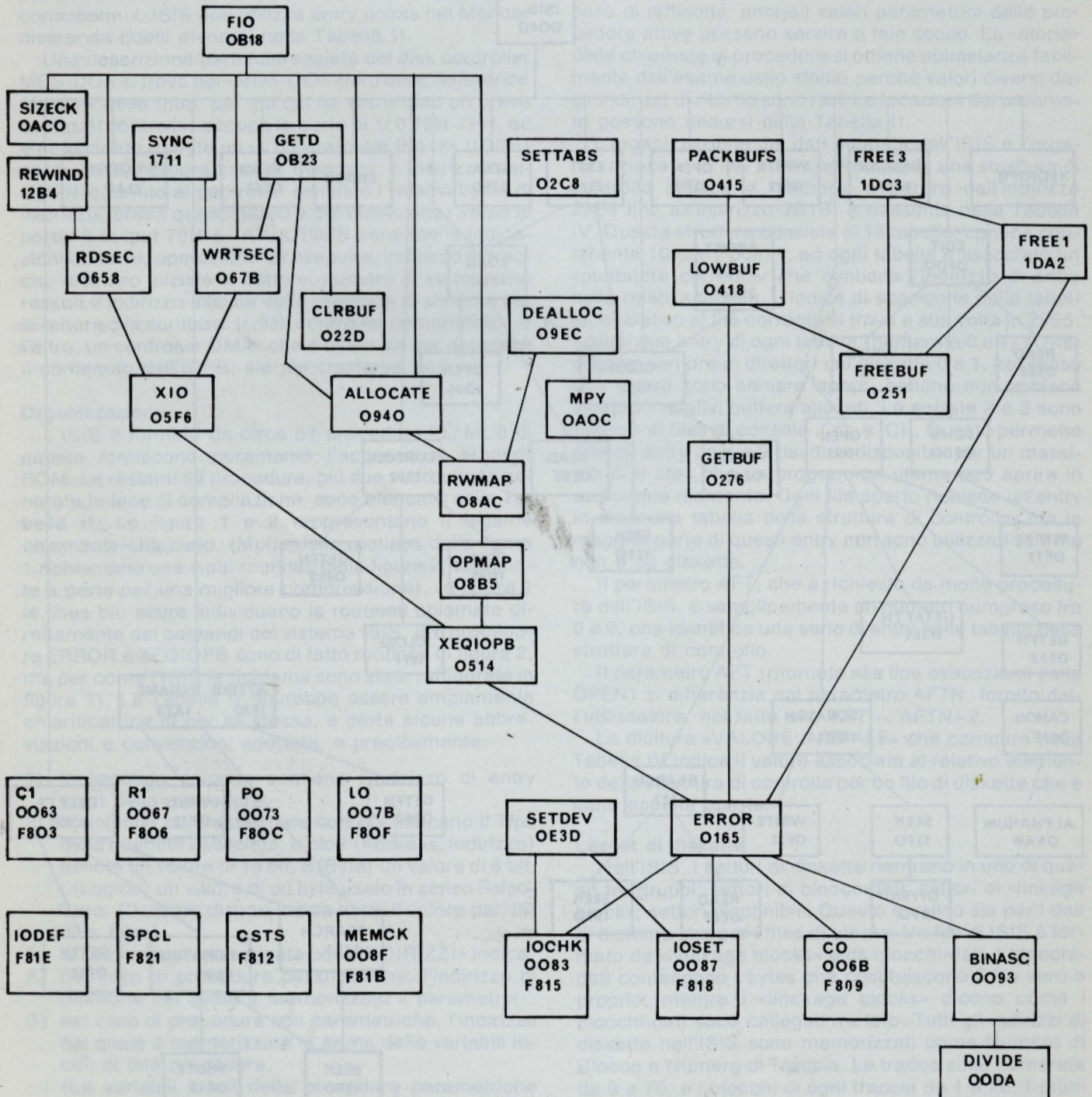
Figura 1 — Procedure di alto livello ISIS.
La combinazione di ATTRIB e RENAME riduce il disordine delle notazioni. Entrambe richiamano gli stessi sottoprogrammi.

direttorio in quanto le relative locazioni gli sono note. Pertanto il primo «linkage block» relativo al direttorio stesso è acquisito dalla Traccia 1, Blocco 1, mentre i blocchi-dati relativi alla Mappa in Bit di Allocazione dei settori sono acquisiti dalla Traccia 2, blocchi 2 e 3. A sua volta il programma di boot-strap in ROM si aspetta di trovare il programma di inizializzazione di sistema a partire dalla Traccia 0, Blocco 1, ed allocato in Traccia 0 per tanti blocchi quanti ne sono necessari; inoltre il programma di inizializzazione di sistema suppone che il primo «linkage block» del file ISIS-BIN (che contiene il programma di sistema vero e proprio) sia in Traccia 2, Blocco 4.

Figura 2 — Basso livello ISIS e procedure di Monitor.

Procedure di sistema «XEQIOPB»

La procedura di sistema XEQIOPB non è descritta nella documentazione ISIS pubblicata. Il codice «di comando» è 68 decimale (44 esadecimale); il blocco-parametri deve contenere i seguenti 3 valori: 534BH, drive, .iopb (dove il drive è 0 o 1 e .iopb è l'indirizzo di memoria dell'IOPB da eseguire). Il suo utilizzo è presumibilmente legato a programmi tipo FORMAT che devono eseguire operazioni non previste dai programmi «normali».

L'IOPB deve essere impostato relativamente al drive 0, indipendentemente da quale drive venga poi specificato, in quanto l'XEQIOPB provvede ad inserire l'identificazione di drive nei posti appropriati. Questa procedura contiene anche il normale trattamento degli errori di diskette. Si osservi, inoltre, che con questa procedura sono saltati gli accessi al direttorio alla mappa dei files e ai flags di protezione, per cui occorre procedere con estrema cautela. Un possibile utilizzo potrebbe essere con un programma di copia diskette, settore per settore. Questo sarebbe molto più veloce del FORMAT $A, dal momento che il meccanismo normale di allocazione del diskette comporta un considerevole lavoro. Inoltre con un tale programma si potrebbe duplicare un diskette non ISIS.

Analoghe procedure esistono nella Versione 1,6 da 16K (in cui l'entrata 534BH del blocco-parametri non è esaminata, ma il drive e l'indirizzo di IOPB sono nondimeno considerati come seconda e terza entrata) e nelle Versione 2,2 dell'ISIS-II (in cui il blocco-parametri contiene solo l'indicazione di drive e l'indirizzo di IOPB, essendo omesso il valore 534BH). Comunque, dal momento che questa è una caratteristica non pubblicizzata dall'Intel, qualsiasi programma che la utilizzi va considerato «system-dependent»; esiste cioè la possibilità che questi programmi non possano lavorare con future versioni dell'ISIS.

Uso delle procedure ISIS nel programma utente

Un programma utente può occasionalmente eseguire un «task» analogo o identico a quello eseguito da una delle procedure ISIS. In tal caso, può essere conveniente richiamare la routine di sistema piuttosto che scrivere e quindi debuggare il codice che esegue *la* stessa funzione. A tale scopo, è necessario conoscere alcuni particolari della implementazione del PL/M.

I parametri di una procedura, fino ad un massimo di due, sono comunicati attraverso i registri. (La prima cosa che fa una procedura parametrica è memorizzare i registri appropriati nelle locazioni di memoria riservate a quei parametri). Una procedura che richiede un parametro aspetterà che il parametro sia nel registro C (se è un byte), o nella coppia di registri BC (se è un indirizzo). Se ci sono due o più parametri, l'ultimo sarà in E (o in DE), e il penultimo in C (o in BC). Il processo «chiamante» deve memorizzare tutti gli altri parametri nelle opportune locazioni di memoria, prima che la procedura sia attivata. (Questo sembra poco elegante, e infatti lo è. I più recenti compilatori del PL/M ammettono il trasferimento dei parametri extra nello stack, il che è molto più bello).

Una procedura che ritorna un valore di byte, lo fornirà nel registro A, mentre il ritorno di un indirizzo avverrà nel registro B (byte più significativo) e nel registro A (byte meno significativo). Questo però non vale per le routines di MPY e di DIVIDE generate al momento della compilazione (vedi Tabella II). Tranne che per i valori ritornati, il contenuto degli altri registri, dopo il rientro da una procedura, in genere, non è attendibile.

È chiaro che i vantaggi dell'impiego di una routine di sistema vanno valutati di contro alla estrema dipendenza dal sistema: le routines di sistema quasi certamente si troveranno in luoghi diversi in differenti versioni del sistema, e inoltre non c'è nessuna garanzia che i loro risultati siano gli stessi di prima.

Conclusione

Gli utilizzatori del sistema operativo ISIS forse sono stati in difficoltà per mancanza di documentazione sul suo funzionamento interno. Anche se tale documentazione non è quasi mai necessaria, in alcuni casi il non averla può diventare un vero problema. Questa presentazione dovrebbe soddisfare in parte questa esigenza degli utilizzatori dei prodotti Intel, e nello stesso tempo aiutare gli hobbysti a capire come è organizzato un piccolo sistema operativo mono-utente.

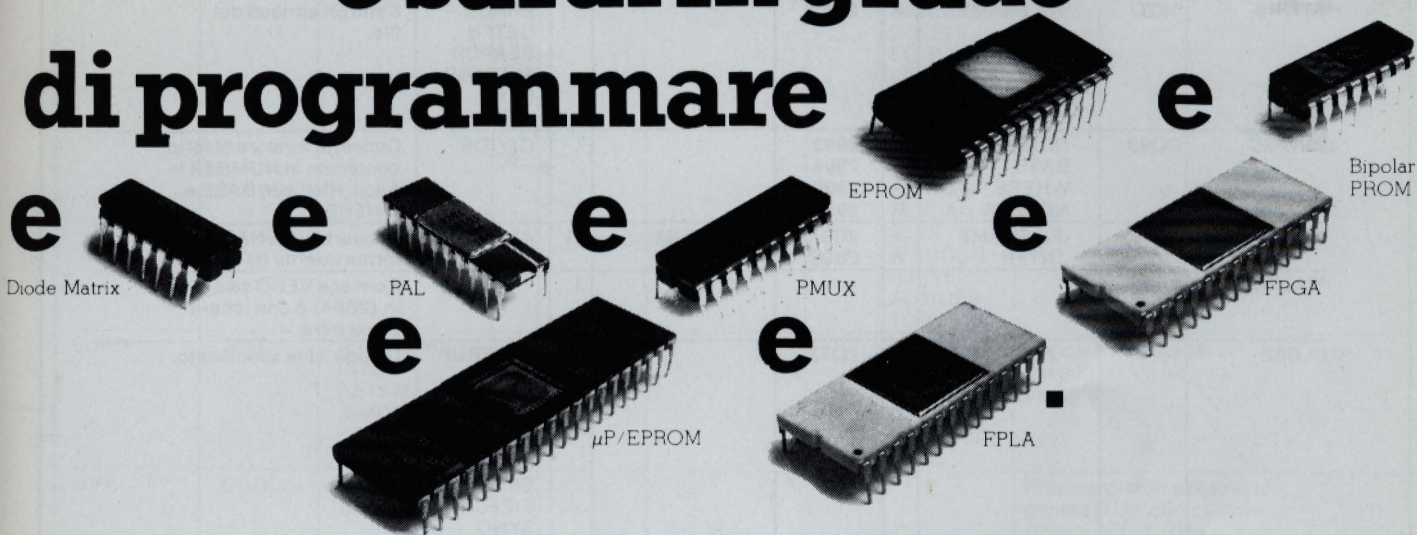
Tabella I — Principali subroutines del monitor di MDS

NOME	ROM ENTRY POINT	ISIS ENTRY POINT	PARAMETRO	T	RITORNI	T	COMMENTO
CI	F8O3	0063			CHARIN	B	Fornisce il carattere letto dal dispositivo di input di console.
CO	F8O9	006B	CHAROUT	B			Uscita del carattere verso il dispositivo di Console.
CSTS	F812	—			STATUS	L	Fornisce VERO se la console ha un carattere pronto.
IOCHK	F815	0083			IOBYTE	B	Fornisce il valore corrente del byte di controllo di I/O.
IODEF	F81E	—	PGMID ENTRY	B A			Definisce l'estensione dell'entry point dell'I/O.
IOSET	F818	0087	IOBYTE	B			Setta il byte di controllo di I/O.
LO	F8OF	007B	CHAROUT	B			Uscita del carattere verso il dispositivo di List.
MEMCK	F81B	008F			MEMSIZE	A	Fornisce il più alto indirizzo di memoria disponibile.
PO	F8OC	0073	CHAROUT	B			Uscita del carattere verso il dispositivo di Punch.
RI	F8O6	0067			CHARIN	B	Fornisce il carattere letto dal dispositivo di Read.
SPCL	F821	—					Entry point riservato per future espansioni.

Acquista il nuovo Sistema 19 e sarai in grado di programmare

EPROM e Bipolar PROM

e Diode Matrix e PAL e PMUX e FPGA

e µP/EPROM e FPLA.

E non dimenticare il nostro nuovo Sistema 17 . . .

Il DATA I/O SISTEMA 17 è la macchina ideale per la produzione, esso offre l'elettronica di programmazione del Sistema 19 con un pannello semplificato per l'uso come duplicatore. Esso realizza il nuovo "Mode Lock" che ti permette di programmare con un solo tasto senza commettere errori.

Se vuoi avere maggiori dettagli su questi nuovi programmatori universali DATA I/O contatta la

SISTREL S.p.A.
Via Timavo 66 - 20099 SESTO S.G. (MI)
tel. (02) 24.85.233 - 24.76.693 Telex 34346
Via G. Armellini 39 - 00143 ROMA
tel. (06) 59.15.551/2/3 Telex 68356

SISTREL
SOCIETÀ ITALIANA STRUMENTI ELETTRONICI S.p.A.

☐ Ricevere un'offerta ☐ Gradirei la visita di un Vs. Tecnico ☐ Essere inseriti nel Vs. mailing list.

NOME .... COGNOME .... TEL. ....

VIA .... CITTA' ....

CAP. .... DITTA .... REPARTO ....

DATA I/O

Programming systems for tomorrow...today

Tabella II — Procedure interne dell'ISIS

NOME	ENTRY POINT	PARAMETRO	T	INDI-RIZZI	RITORNI	T	SUBROUT.	COMMENTO
ABORT	2584			2DFF			CLOSE XEQIOPB	Termina l'esecuzione e reinizializza il sistema; l'entry usuale avviene tramite l'interrupt vettorizzato 1 all'indirizzo 0008.
ALLOCATE	O94O	DRIVE	B	2CC2	TRKSEC	A	ERROR RWMAP	Alloca un nuovo blocco-diskette e ritorna il relativo indirizzo.
ATTRIB	187O	.FILENAME	A	2D64			ERROR GETFN SEARCH SETTABS SYNC WRITE	Setta gli attributi del file.
BINASC	OO93	NUMBER BASE WHERE NDIGITS	A A A B	2992 2994 2996 2998			DIVIDE	Converte il valore binario contenuto in NUMBER in Ascii. Richiede BASE e WHERE.
CANON	O69E	.FILENAME .INTFN	A A	2B9A 2B9C	ERRNUM	B	ALPHANUM	Converte FILENAME nella forma interna INTFN.
**ALPHANUM	6AB			——	LETDIG	L		Fornisce VERO se il byte in (2B94) è una lettera o una cifra.
CLOSE	18FF	AFT	B	2D73			FREEBUF MPY READ REWIND RWMAP SEEK SETTABS SIZECK SYNC WRITE WRTSEC	Chiude il file specificato.
CLRBUF	O22D	.BUFFER	A	29DE				Cancella i 128 bytes di BUFFER riempiendoli di 0.
CONSOLE	2182	.INFILE .OUTFILE	A A	2DDO 2DD2			CLOSE ERROR GETFN OPEN	Cambia le assegnazioni del file di console.
DEALLOC	O9EC	DRIVE TRKSEC	B A	2CC9 2CCA			MPY RWMAP	Rilascia un blocco-diskette.
DELETE	1CCB	.FILENAME	A	2D9E			ERROR GETFN SCRATCH SEARCH SETTABS	Cancella il file specificato.
DIVIDE	OODA	(BC) (DE)	A A	 ——	(BC) (DE)	A A		Routine di divisione generata dal compilatore. Divide BC per DE; rientra con il quoziente in BC e il resto in DE.
EDIT	OFDO	AFT .BUFFER COUNT .ACTUAL	B A A A	2D11 2D12 2D14 2D16			ECHO ERROR READ WRITE	È chiamata al posto di READ quando è richiesto il «line-editing».
**ECHO	OFDD	CHAR	B	2D28			SETTABS WRITE	Scrive CHAR su un «echo» file.
ERROR	O165	FLAG CODE	B B	29A2 29A4			BINASC CO IOCHK IOSET	Se FLAG=0, ritorna CODE all'utente. Se FLAG=1, scrive il messaggio. Se FLAG=2, scrive il messaggio e va in abort.
EXIT	252A			2DFE			CLOSE LOAD RWMAP	Chiude i files e ritorna al sistema.
FIO	OB18	AFT .BUFFER COUNT .ACTUAL READFLAG	B A A A L	2CCF 2CDO 2CD2 2CD4 2CD6			ALLOCATE CLRBUF GETD RDSEC RWMAP SIZECK WRTSEC	Trasferisce i bytes (in numero dato da COUNT) dal buffer del file di diskette a BUFFER chiamante: READFLAG è VERO per la lettura del file.
**GETD	OB23			——			ALLOCATE CLRBUF WRTSEC	Alloca nuovi blocchi-dati, come richiesto.

NOME	ENTRY POINT	PARAMETRO	T	INDI-RIZZI	RITORNI	T	SUBROUT.	COMMENTO
FREEBUF	0251	.BUFFER	A	29E2				Rilascia BUFFER al buffer pool.
GETAFT	03E7			2B1E	AFT	B	ERROR	Assegna AFT al file da aprire.
GETBUF	0276			29E4	.BUFFER	A	ERROR	Alloca un buffer e ritorna il relativo indirizzo.
GETFN	0882	.FILENAME .INTFN	A A	2BA4 2BA6			CANON ERROR	Passa i parametri a CANON; chiama ERROR se necessario.
ISIS	0040	COMMAND BLOCK	B A	2921 2922			ATTRIB CLOSE CONSOLE DELETE EDIT ERROR EXIT LOAD OPEN READ RENAME RESCAN SEEK WHOCON WRITE XEQIOPB ?OPEN	Principale entry point utente per il sistema operativo.
LOAD	22EC	.FILENAME BIAS RETSW .ENTRY	A A A A	2DE8 2DEA 2DEC 2DEE			CLOSE ERROR MEMCK OPEN PACKBUFS READ	Carica e in opzione esegue un file binario assoluto.
MPY	OAOC	(BC) (DE)	A A	— —	(DE)	A		Routine di moltiplicazione generata dal compilatore. (DE) = (BC)* (DE)
OPEN	1D95	.FILENAME ACCESS ECHOAFT	A A A	2DA2 2DA4 2DA6	AFTN	B	ALLOCATE CLRBUF ERROR FREE1 FREE3 GETAFT GETBUF GETFN REWIND RWMAP SCRATCH SEARCH SETTABS SYNC WRITE WRTSEC	Apre il file specificato; assegna e ripone l'indice nella tabella del file.
** FREE1	1DA2	ERRNUM	B	2DBO			ERROR FREEBUF	Libera il buffer di«line-edit» (se ne esiste qualcuno), poi chiama ERROR (0, ERRNUM).
** FREE3	1DC3	ERRNUM	B	2DB1			FREEBUF FREE1	Libera i buffers di «link» e di dati, poi chiama FREE1 (ERRNUM).
PACKBUFS	0415			2B2O	LOWLIM	A	LOWBUF	Sposta i buffers utilizzati nella più bassa locazione possibile; ritorna il più basso indirizzo libero.
** LOWBUF	0418	.BUFPTR	A	2B22			FREEBUF GETBUF	Sposta il buffer il cui indirizzo è in BUFFER in una posizione più bassa, se è disponibile.
RDSEC	0658	TRKSEC .BUFFER	A A	2B44 2B46			X10	Legge il blocco-diskette richiesto, in BUFFER.
READ	0E63	AFT .BUFFER COUNT .ACTUAL	B A A A	2CDD 2CDE 2CEO 2CE2			CI ERROR FIO RI	Legge caratteri dal file in BUFFER, fino a che COUNT è soddisfatto o è raggiunto l'«end of file».
RENAME	1AE8	.OLDNAME .NEWNAME	A A	2D8E 2D9O			SETDEV SETTABS ERROR GEFTN SEARCH SETTABS	Cambia il nome di un file.

(segue)

(continua da pagina precedente)

NOME	ENTRY POINT	PARAMETRO	T	INDI-RIZZI	RITORNI	T	SUBROUT.	COMMENTO
							SYNC WRITE	
RESCAN	1276	AFT	B	2D29			ERROR SETTABS	Setta il puntatore di byte-successivo di un file «editato» per linee, all'inizio della linea corrente.
REWIND	12B4			—			RDSEC	Inizializza alcune tabelle di file e legge il primo «link block».
RWMAP	08AC	DRIVE OPCD	B B	2CC0 2CC1			OPMAP	Legge o scrive la «bit-map» del drive richiesto, usando il buffer interno.
**OPMAP	08B5			—			XEQIOPB	Costruisce l'IOPB per RWMAP.
SCRATCH	1BB7	DRIVE	B	2D95			DEALLOC RWMAP SEEK SETTABS WRITE XEQIOPB	Cancella tutti gli spazi di diskette assegnati al file il cui entry di direttorio è in 2D4A.
SEARCH	1723	.INTFN	A	2D5C			FREEBUF GETBUF MPY READ REWIND SEEK SETTABS	Cerca il file rappresentato da INTFN nel direttorio appropriato.
SEEK	12F0	AFT MODE .BLOCKNO .BYTENO	B A A A	2D2D 2D2E 2D30 2D32			ALLOCATE CLRBUF ERROR OFFEND RDSEC REWIND RWMAP SETTABS SIZECK WRTSEC	Riposiziona il puntatore di byte-successivo di un file di diskette.
**OFFEND	12FD	BLKNUMA BLKNUMB BYTENUMA BYTENUMB	A A B B	2D44 2D46 2D48 2D49	LARGER	L		Ritorna VERO se l'operazione di «seek» tende ad estendere il file.
SETDEV	0E3D			—			IOCHK IOSET	Setta «IOBYT» di ROM per eseguire l'I/O correttamente.
SETTABS	02C8	AFT	B	2B1D				Setta i puntatori correnti nelle tabelle di file.
SIZECK	0AC0			—				Aggiorna la lunghezza di file nelle tabelle.
SYNC	1711			—			WRTSEC	Forza il ritardo di scrittura del blocco-dati corrente.
WHOCON	22A0	AFT .BUFFER	B A	2DE1 2DE2				Copia in BUFFER il nome del file di console richiesto.
WRITE	0F12	AFT .BUFFER COUNT	B A A	2CF7 2CF8 2CFA			CO ERROR FIO LO PO SETDEV SETTABS	Scrive i caratteri (di n. COUNT) nel file da BUFFER.
WRTSEC	067B	TRKSEC .BUFFER	A A	2B48 2B4A			XIO	Scrive il blocco specificato, da BUFFER sullo opportuno drive, per il file corrente.
XEQIOPB	0514	DRIVE .IOPB	B A	2B29 2B2A			ERROR	È eseguita fornendo IOPB sul drive richiesto.
XIQ	05F6	OPCODE DRIVE TRKSEC .BUFFER	B B A A	2B34 2B35 2B36 2B38			XEQIOPB	Costruisce IOPB per eseguire l'operazione richiesta.
?OPEN	25B8			—			ERROR	Incrementa di 2 il valore di AFTN, fornito dall'utente, per ottenere il valore interno; se questo non corrisponde ad un file aperto, chiama ERROR (0, 2).

Tabella III — Variabili di significato diagnostico dell'ISIS

INDIRIZZI	CONTENUTI
28CO-2920	Area non usata, riservata per espansioni future. Può essere usata per il codice provvisorio del debugging, o per «patches».
2928-2931	Copia del corrente blocco parametri.
294C-294D	«Stack Pointer» dell'utente.
294E-298D	Stack ISIS.
299C	Commutatore debug (logico). Quando è Vero, in caso di errori «fatali» sarà abilitato il monitor di MDS invece di abortire il programma.
299D	Flag comando «CONSOLE». Il valore è 2 finchè è in corso una procedura di CONSOLE, e 0 in tutti gli altri casi. Questo rende «fatali» tutti gli errori rilevati da CONSOLE.
299E-299F	«Byte Risultato» e «Type Risultato» generati dal canale di disk per un errore di disk non recuperabile. (Questi bytes sono stampati come parte del messaggio di «ERRORE 24»).
29CB-29DD	Tabella buffer, contenente un byte per ciascuno dei 19 possibili buffers. I possibili valori di ciascun byte sono: 0 Libero. 1 Spazio riservato da LOAD in quanto il buffer contiene una parte del programma utente. 2 Usato come buffer. I buffers iniziano all'indirizzo 2E00; ciascun buffer è lungo 128 bytes.
29E5-2B18	Struttura file (vedi Tabella IV).
2B29	Drive di diskette pilotato per ultimo.
2B2A-2B2B	Indirizzo di IOPB.
2B2C	Byte risultato.
2B2D	Tipo risultato.
2B4C-2B75	Tabella dei nomi di device (ogni nome occupa 2 byte).
2B86-2B8F	Nome del formato interno dell'ultimo file cercato da SEARCH. 2B86 contiene l'identificazione del device (l'indice è nella Tabella in 2B4C), 2B87-2B8C contengono il nome del file, 2B8D-2B8F contengono l'estensione del nome.
2BA8-2CA7	Buffer della «Bit Map».
2CA8	Nome del drive a cui appartiene la «Bit Map».
2CA9	(Logico) Vero se la «bit map» è stata modificata da quando fu scritta.
2D4A-2D59	Entry di direttorio relativo all'ultimo file trovato da SEARCH. (Questo non è aggiornato con la crescita del file).
2DB2-2DCO	Nome del file di :CI: corrente.
2DC1-2DCF	Nome del file di :CO: corrente.
2DF6-2DF7	Indirizzo dell'ultimo record logico letto da LOAD. (Eccetto che durante LOAD, questo torna ad essere l'entry point dell'ultimo file caricato).
2EOO—	Area buffer.

Tabella IV — Struttura di controllo dei files ISIS (in memoria)

NOME	INDIRIZZO DI BASE	T	PUNTATORE CORRENTE	VALORE INIZIALE	DESCRIZIONE
CLOSED	2AOE	L	29EA		Vero se questo file non è aperto.
DEVICE	2A18	B	29EC		Identificatore di device (vedi Tabella V).
ACCESS	2A22	B	29EE		Valore del parametro ACCESS fornito in OPEN.
ECHOAFT	2A2C	B	29FO		AFT di «echo» file; 0 se non-edit.
EBUF	2A36	A	29F2		Indirizzo del buffer di edit.
DBUF	2A4A	A	29F4		Indirizzo del buffer-dati (copia dell'elemento corrente in 29E8).
BYTENO	2A5E	B	29F6	128	Contatore dei bytes contenuti nel buffer-dati.
DNUM	2A68	A	29F8		Posizione nel direttorio.
LBUF	2A7C	A	29FA		Indirizzo del buffer di «link block» (copia dell'elemento corrente in 29E6).
DPTR	2A9O	B	29FC	1	Puntatore di word nel «link block».
LASTBYTE	2A9A	B	29FE		Numero dei bytes nell'ultimo blocco-dati.
ALLOC	2AA4	L	2AOO	F	Vero se l'allocazione è stata fatta per il file.
DMOD	2AAE	L	2AO2	F	Vero se il blocco-dati corrente è stato modificato.
BLKCNT	2AB8	A	2AO4		Numero di blocchi-dati nel file.
BLKNO	2ACC	A	2AO6	O	Numero sequenziale del blocco-dati corrente del file.
LADDR	2AEO	A	2AO8	(2AF4)	Indirizzo di traccia-settore del «link block» corrente.
L1ADDR	2AF4	A	2AO4		Indirizzo di traccia-settore del primo «link block» del file.
DADDR	2B08	A	2AOC	O	Indirizzo di traccia-settore del blocco-dati corrente.

Tabella V

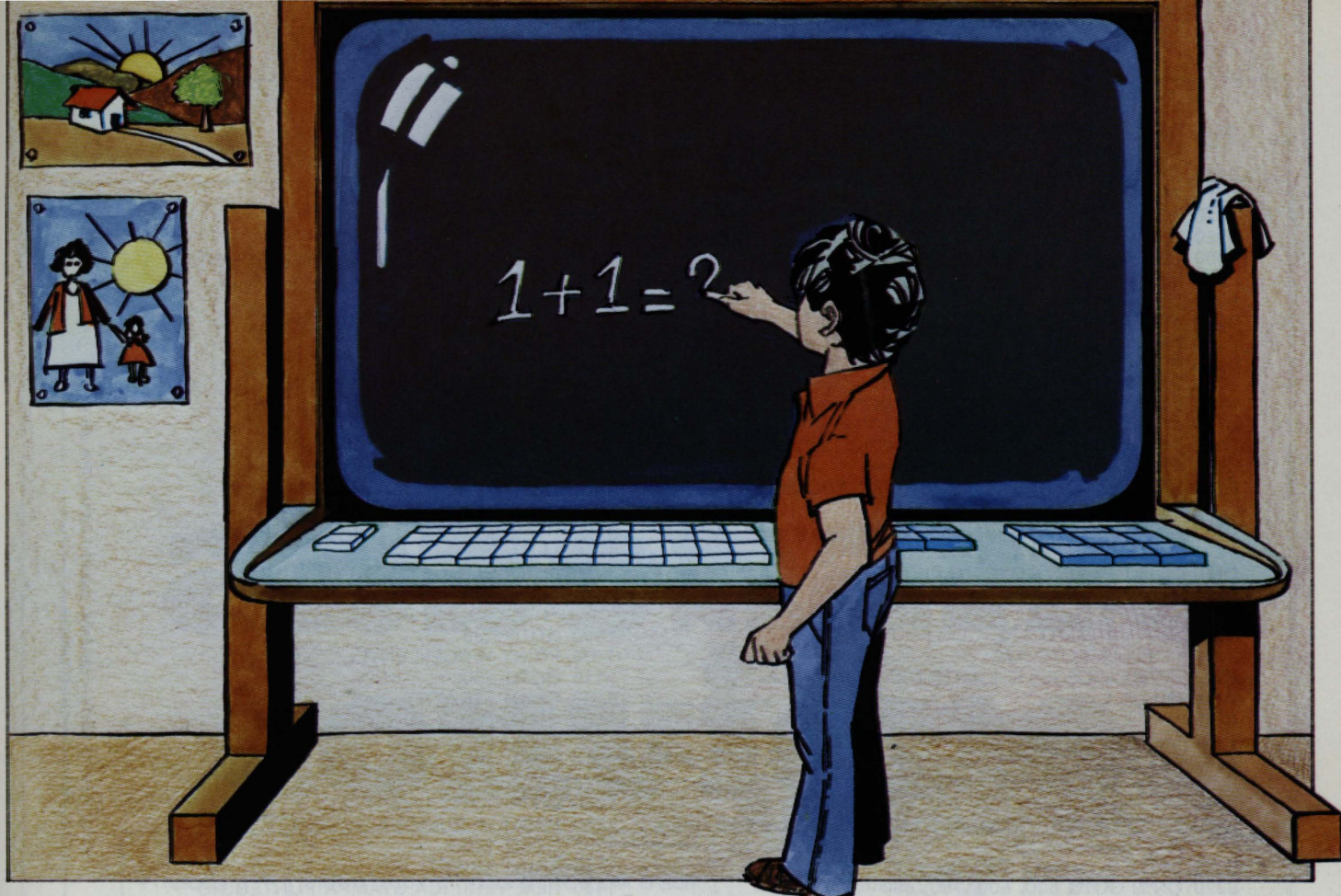
Codici di identificazione device in ISIS

NUMERO	DEVICE
0	:FO:
1	:F1:
2	:TI:
3	:TO:
4	:VI:
5	:VO:
6	:I1:
7	:01:
8	:TR:
9	:HR:
10	:R1:
11	:R2:
12	:TP:
13	:HP:
14	:P1:
15	:P2:
16	:LP:
17	:L1:

Tabella VII. Entry di direttorio in ISIS

BYTES	CONTENUTI
00	Falg: 00 = attivo 7F = mai usato FF = cancellato
01-06	Nome file
07-09	Estensione nome
0A	Attributi: Bit 0 = trasparente Bit 1 = sistema Bit 2 = protetto in scrittura Bit 7 = format
0B	Numero dei bytes nell'ultimo blocco-dati
0C-0D	Numero dei blocchi-dati
0E-0F	Indirizzo (Blocco, Traccia) del primo «linkage block»

Tabella VI. Formato files in ISIS

Nome del File	Link	Dati	Contenuti
ISIS.TO	0,18	0,1—0,17	Programma di inizializzazione sistem.
ISIS.LAB	0,19	0,1A	Etichetta di diskette.
ISIS.DIR	1,1	1,2—1,1A	Direttorio di diskette (vedi Tabella VII).
ISIS.MAP	2,1	2,2—2,3	Allocazione mappa settori («bit-map»).

Matematica elementare

di Mike Donahue

È riportato un intero programma in BASIC di facile comprensione. È un esempio di programmi di auto-apprendimento che mostra una delle possibili utilizzazioni del personal computer.

Se avete bambini in famiglia, o avete a che fare con bambini nel vostro lavoro, questo può essere un programma pratico per voi. Scritto in SWTPC 4K BASIC, è essenzialmente un programma che aiuta i bambini ad imparare i fondamenti della matematica. Il programma, pur non essendo destinato unicamente ai bambini delle scuole elementari, fornisce esercizi che utilizzano tutte e quattro le funzioni matematiche fondamentali ed opera su numeri massimo di quattro cifre.

È un programma molto divertente, di tipo colloquiale, e per di più dà conferma immediata alle proprie risposte. In sintesi, il programma permette di affrontare svariati problemi matematici, facili e difficili.

Il tipo di problema è scelto da un elenco che si può chiedere o meno di stampare (visualizzare) al momento in cui il programma inizia l'esecuzione. Oltre alle quattro funzioni matematiche fondamentali (addizione, sottrazione, moltiplicazione e divisione), è possibile realizzare una loro combinazione: ad esempio si possono richiedere insieme le operazioni di addizione e sottrazione, oppure di moltiplicazione e divisione, o infine di tutte e quattro le funzioni in ordine casuale.

Il programma va adattato al livello specifico di conoscenze matematiche del bambino; per questo la complessità dei problemi è variabile ed è controllata mediante il numero che viene impostato come «limite del problema». Quanto più è grande il numero che si imposta, tanto più sarà difficile il problema, e viceversa. Ad esempio, impostando «5», i numeri usati nel problema vanno da 1 a 5. Questo limite vale per tutte le operazioni matematiche tranne la divisione, in cui per semplicità i problemi sono stati progettati in modo da non dare resto (Il metodo scelto per garantire ciò consiste nel moltiplicare uno dei numeri dell'esercizio per l'altro). Perciò il limite superiore dei numeri nei problemi di divisione sarà maggiore del valore impostato come «limite del problema».

Il programma utilizza l'istruzione di generazione casuale di numeri per i valori da inserire nei problemi matematici, per le combinazioni dei tipi di problemi e per avere una certa varietà di messaggi di risposta esatta o errata.

Se si modifica questo programma per un altro BASIC, bisogna verificare l'istruzione di generazione casuale di numeri.

Il punteggio è stampato dopo ogni serie di 10 problemi. Al termine dei 50 problemi, il gioco termina automaticamente.

Listing del Programma

```
0010 REM PROGRAMMA MATEMATICO PER ADDIZIO-
     NE E SOTTRAZIONE
0015 REM PROGRAMMA MATEMATICO PER MOLTIPLICA-
     ZIONE E DIVISIONE
0020 PRINT «******** TESTO MATEMATICO********»
0050 LET E=0
0060 LET G=0
0070 LET E1=0
0080 LET G1=0
0082 PRINT «IMPOSTA 1 PER TABELLA PROBLEMI»
0084 INPUT I
0086 IF I <> 1 GOTO 170
0090 PRINT «I PROBLEMI DISPONIBILI SONO:»
0100 PRINT «1 PER SOLE ADDIZIONI»
0110 PRINT «2 PER SOLE SOTTRAZIONI»
0120 PRINT «3 PER SOLE MOLTIPLICAZIONI»
0130 PRINT «4 PER SOLE DIVISIONI»
0140 PRINT «5 PER COMBINAZIONI DI ADDIZIONE E SOT-
     TRAZIONE»
0150 PRINT «6 PER COMBINAZIONI DI MOLTIPLICAZIONE E
     DIVISIONE»
0160 PRINT «7 PER COMBINAZIONI DI TUTTE E 4»
0170 PRINT «SCEGLI»
0175 INPUT P
0180 IF P <1 GOTO 170
0190 IF P > 7 GOTO 170
0200 PRINT «SCEGLI IL LIMITE DEL PROBLEMA»
0210 INPUT L
0220 IF L <1 GOTO 200
0230 LET A = INT (L* RND (0) + 1)
0240 LET B = INT (L* RND (0) + 1)
0250 IF G+E = 10 GOSUB 1500
0260 ON P GOTO 500, 600, 700, 800, 400, 450, 300
0300 REM COMBINAZIONE CASUALE DI TUTTE E 4 LE OPE-
     RAZIONI
0310 LET S = INT (4* RND (0) + 1)
0320 ON S GOTO 500, 600, 700, 800
0400 REM COMBINAZIONE CASUALE DI ADDIZIONE E SOT-
     TRAZIONE
0410 LET S = INT (2* RND (0) + 1)
0420 ON S GOTO 500, 600
0450 REM COMBINAZIONE CASUALE DI MOLTIPLICAZIONE
     E DIVISIONE
0460 LET S = INT (2* RND (0) + 1)
0470 ON S GOTO 700, 800
0500 REM ROUTINE DI ADDIZIONE
0510 PRINT A; «+»; B; «=»;
0520 INPUT C
0530 IF C = A+B GOTO 1000
0540 GOSUB 1200
0550 GOTO 510
0600 REM ROUTINE DI SOTTRAZIONE
0610 IF A>B GOTO 660
0620 LET C=A
0630 LET A=B
0640 LET B=C
0660 PRINT A; «—»; B; «=»;
0670 INPUT C
0680 IF C = A—B GOTO 1000
0685 GOSUB 1200
0690 GOTO 660
0700 REM ROUTINE DI MOLTIPLICAZIONE
0710 PRINT A; «×»; B; «=»;
0720 INPUT C
0730 IF C = A* B GOTO 1000
0740 GOSUB 1200
0750 GOTO 710
0800 REM ROUTINE DI DIVISIONE
0810 LET A = A* B
0820 PRINT A; «/»; B; «=»;
0830 INPUT C
0840 IF C = A/B GOTO 1000
0850 GOSUB 1200
0860 GOTO 820
1000 REM ROUTINE DI RISPOSTA ESATTA
1010 LET G = G + I
1020 LET R = INT (3* RND (0) +1)
1030 ON R GOTO 1080, 1100, 1120
1080 PRINT «OTTIMO»
1090 GOTO 230
1100 PRINT «ECCELLENTE»
1110 GOTO 230
1120 PRINT «ESATTO»
1130 GOTO 230
1200 REM ROUTINE DI RISPOSTA ERRATA
1210 LET E = E+1
1220 LET R = INT (2* RND (0) + 1)
1230 ON R GOTO 1280, 1320
1280 PRINT «* RIPROVA..»;
1290 GOTO 1400
1320 PRINT «RIPETI..»;
1400 REM VERIFICA STAMPA PUNTEGGIO
1410 IF G+E <> 10 RETURN
1420 GOSUB 1500
1430 PRINT «NON DIMENTICARE QUESTO!»;
1440 RETURN
1500 REM ROUTINE DI STAMPA PUNTEGGIO
1510 LET E1 = E1 + E
1520 LET G1 = G1 + G
1530 LET E = 0
1540 LET G = 0
1550 LET T = G1 + E1
1555 PRINT
1560 PRINT «IL TUO PUNTEGGIO AL TERMINE DI ";T;" PRO-
     BLEMI E'»
1570 PRINT G1; «ESATTI E»; E1; «ERRATI»
1575 IF T >50 RETURN
1700 REM ROUTINE DI STOP
1710 PRINT «GRAZIE PER IL GIOCO!»
9999 END
```

Nota: le virgolette « devono essere sostituite da apici " per la corretta esecuzione del programma.

Esecuzione del Programma

```
Run
******** TESTO MATEMATICO********
IMPOSTA 1 PER TABELLA PROBLEMI? 1
I PROBLEMI DISPONIBILI SONO:
  1 PER SOLE ADDIZIONI
  2 PER SOLE SOTTRAZIONI
  3 PER SOLE MOLTIPLICAZIONI
  4 PER SOLE DIVISIONI
  5 PER COMBINAZIONI DI ADDIZIONE E SOTTRAZIONE
  6 PER COMBINAZIONI DI MOLTIPLICAZIONE E DIVI-
    SIONE
  7 PER COMBINAZIONI DI TUTTE E 4
SCEGLI? 7
SCEGLI IL LIMITE DEL PROBLEMA? 10

    7 — 6 = ? 1          6 + 8 = ? 14
    ECCELLENTE           ECCELLENTE
    2 + 3 = ? 5          3 × 10 = ? 30
    ESATTO               OTTIMO
    3 + 1 = ? 4          7 — 3 = ? 4
    ESATTO               ECCELLENTE

    24 / 3 = ? 8         1 + 3 = ? 4
    ESATTO               ECCELLENTE
    9 + 8 = ? 17         6 + 9 = ? 3
    ECCELLENTE           * RIPROVA

IL TUO PUNTEGGIO AL TERMINE DI 10 PROBLEMI
È 9 ESATTI E 1 ERRATO.
NON DIMENTICARE QUESTO!

    6 + 9 = ? 15         6 — 6 = ? 0
    ECCELLENTE           OTTIMO
    1 × 3 = ? 1          7 — 1 = ? 6
    * RIPETI..           OTTIMO
    1 × 3 = ? 3          27 / 9 = ? 3
    ESATTO               ESATTO
    ....................
    ....................
    ....................
IL TUO PUNTEGGIO AL TERMINE DI 50 PROBLEMI
È 45 ESATTI E 5 ERRATI.
GRAZIE PER IL GIOCO!
```

SIEMENS

SKC 85

L'SKC 85 è un microcomputer completo su un modulo formato europeo 100 x 160 mm.
Realizzato con il microprocessore SAB 8085, dispone di 4 KByte di memoria di programma (ROM/EPROM), di 768 Byte di RAM (di cui 256 Byte CMOS con batteria tampone) di 4 ingressi di interrupt e di 46 linee di ingresso/uscita.
Dispone ancora di 2 contatori-timer a 14 bit e lavora con una sola tensione di alimentazione (+5 V).

In posizione intermedia tra i microcomputer su un solo chip ed i sistemi modulari di tipo universale è particolarmente adatto al controllo di macchine automatiche di media complessità.

SIEMENS ELETTRA S.P.A.
Divisione componenti apparecchi e sistemi di misura - Reparto A 201
20124 Milano - Via Fabio Filzi 25/A - tel. (02) 6248

un single board computer per applicazioni di controllo